Face Detection and Modeling for Recognition

By

Rein-Lien Hsu

A Dissertation

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science & Engineering

2002

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2002		**00-00-2002 to 00-00-2002**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Face Detection and Modeling for Recognition	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Michigan State University,Department of Computer Science & Engineering,East Lansing,MI,48824**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**198**	

Abstract

Face Detection and Modeling for Recognition

By

Rein-Lien Hsu

Face recognition has received substantial attention from researchers in biometrics, computer vision, pattern recognition, and cognitive psychology communities because of the increased attention being devoted to security, man-machine communication, content-based image retrieval, and image/video coding. We have proposed two automated recognition paradigms to advance face recognition technology. Three major tasks involved in face recognition systems are: (i) face detection, (ii) face modeling, and (iii) face matching. We have developed a face detection algorithm for color images in the presence of various lighting conditions as well as complex backgrounds. Our detection method first corrects the color bias by a lighting compensation technique that automatically estimates the parameters of reference white for color correction. We overcame the difficulty of detecting the low-luma and high-luma skin tones by applying a nonlinear transformation to the YC_bC_r color space. Our method generates face candidates based on the spatial arrangement of detected skin patches. We constructed eye, mouth, and face boundary maps to verify each face candidate. Ex-

perimental results demonstrate successful detection of faces with different sizes, color, position, scale, orientation, 3D pose, and expression in several photo collections.

3D human face models augment the appearance-based face recognition approaches to assist face recognition under the illumination and head pose variations. For the two proposed recognition paradigms, we have designed two methods for modeling human faces based on (i) a generic 3D face model and an individual's facial measurements of shape and texture captured in the frontal view, and (ii) alignment of a semantic face graph, derived from a generic 3D face model, onto a frontal face image. Our modeling methods adapt recognition-oriented facial features of a generic model to those extracted from facial measurements in a global-to-local fashion. The first modeling method uses displacement propagation and 2.5D snakes for model alignment. The resulting 3D face model is visually similar to the true face, and proves to be quite useful for recognizing non-frontal views based on an appearance-based recognition algorithm. The second modeling method uses interacting snakes for graph alignment. A successful interaction of snakes (associated with eyes, mouth, nose, etc.) results in appropriate component weights based on distinctiveness and visibility of individual facial components. After alignment, facial components are transformed to a feature space and weighted for semantic face matching. The semantic face graph facilitates face matching based on selected components, and effective 3D model updating based on 2D images. The results of face matching demonstrate that the proposed model can lead to classification and visualization (e.g., the generation of cartoon faces and facial caricatures) of human faces using the derived semantic face graphs.

To my parents; my lovely wife, Pei-Jing; and my son, Alan

Acknowledgments

First of all, I would like to thank all the individuals who have helped me during my Ph.D. study at Michigan State University. I would like to express my deepest gratitude to my advisor, Dr. Anil K. Jain, for his guidance in academic research and his support in daily life. He broadened my view in research areas, especially in pattern recognition and computer vision, and taught me how to focus on research problems. I will never forget his advice "Just do it," while being caught in multiple tasks at the same time. I am grateful to my Ph.D. committee, Dr. Mohamed Abdel-Mottaleb, Dr. George Stockman, Dr. John J. Weng, and Dr. Sarat C. Dass, for their valuable ideas, suggestions, and encouragement.

I would also like to thank Dr. Chaur-Chin Chen and Dr. Wey-Shiuan Hwang for their help at the beginning of my study at MSU, and Dr. Shaoyun Chen and Yonghong Li for their help in the NASA modeling project. I am very grateful to Dr. Helen Shen and Dr. Mihran Tuceryan for their numerous suggestions and discussions on model compression. Special thanks are due to Philips Research-USA for offering me summer internships in 2000 and 2001; to Dr. Mohamed Abdel-Mottaleb for his guidance and suggestions in my work on face detection; to Dr. Patrick Flynn

for providing the range datasets; to Dr. Wey-Shiuan Hwang for providing his face recognition software; and to Dennis Bond for his help in creating a graphical user interface for face editing.

Thanks are also due to Cathy M. Davison, Linda Moore, Starr Portice, and Beverly J. Wallace for their assistance in the administrative tasks. Special thanks to all the Prippies: Lin Hong, Aditya Vailaya, Nicolae Duta, Salil Prabhakar, Dan Gutchess, Paul Albee, Arun Ross, Anoop Namboodiri, Silviu Minut, Umut Uludag, Xiaoguang Lu, Martin Law, Miguel Figueroa-Villanue, and Yilu Zhang for their help during my stay in the PRIP Lab in the Department of Computer Science and Engineering at MSU. I would also like to thank Michael E. Farmer for giving me an opportunity to work on human tracking research.

I would like to thank Mark H. McCullen for mentoring me to be a teaching assistant in CSE232; Dr. Jeffrey A. Fessler at the University of Michigan, Ann Arbor, for his valuable help during my transfer to the Dept. of Computer Science at Michigan State University; and Dr. Yung-Nien Sun and Dr. Chin-Hsing Chen in Taiwan for their encouragement and spiritual support. Special thanks to NASA, Philips Research-USA, Eaton corporation, and ONR (grant NO. N00014-01-1-0266) for their financial support during my Ph.D studies.

Finally, but not the least, I would like to thank my parents, my wife, Dr. Pei-jing Li, and my son, Alan, for all the happiness they have shared with me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

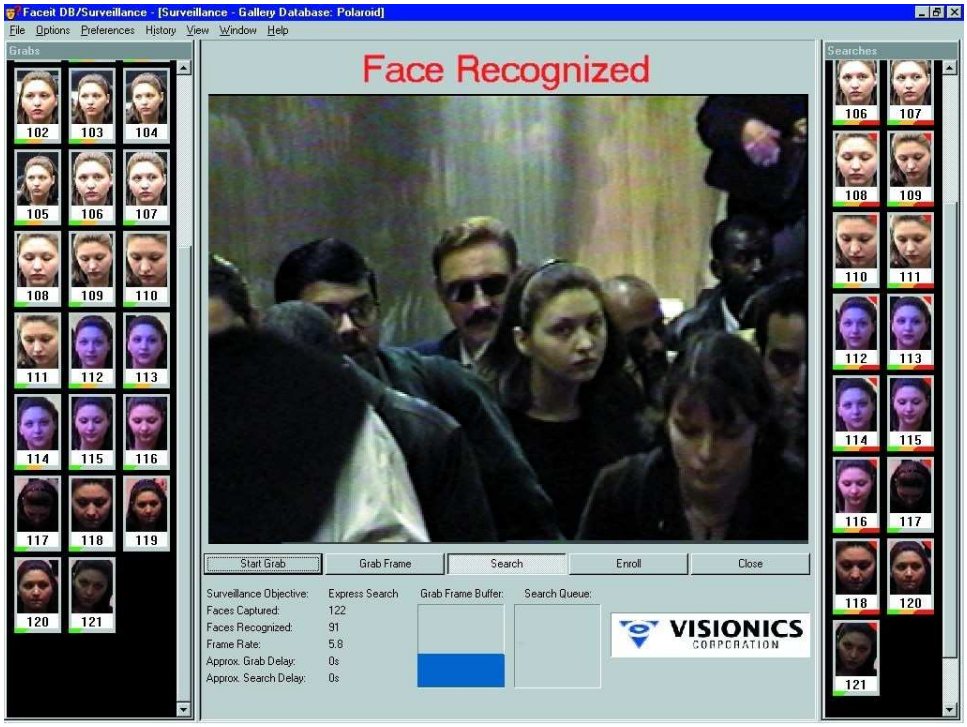
In recent years face recognition has received substantial attention from researchers in biometrics, pattern recognition, and computer vision communities (see surveys in [36], [37], [38]). This common interest among researchers working in diverse fields is motivated by our remarkable ability to recognize people (although in case of certain rare brain disability, e.g., prosopagnosia or face blindness [39], this recognition ability is lost) and the fact that human activity is a primary concern both in everyday life and in cyberspace. Besides, there are a large number of commercial, security, and forensic applications requiring the use of face recognition technology. These applications (see Fig. 1.1) include automated video surveillance (e.g., super bowl face scans and airport security checkpoints), access control (e.g., to personal computers and private buildings), mugshot identification (e.g., for issuing driver licenses), design of human computer interface (HCI) (e.g., classifying the activity of a vehicle driver), multimedia communication (e.g., generation of synthetic faces), and content-based image database management [40]. These applications involve locating, tracking, and

recognizing a single (or multiple) human subject(s) or face(s).

Face recognition is an important biometric identification technology. Facial scan is an effective biometric attribute/indicator. Different biometric indicators are suited for different kinds of identification applications due to their variations in intrusiveness, accuracy, cost, and (sensing) effort [5] (see Fig. 1.2(a)). Among the six biometric indicators considered in [6], facial features scored the highest compatibility, shown in Fig. 1.2(b), in a machine readable travel documents (MRTD) system based on a number of evaluation factors, such as enrollment, renewal, machine requirements, and public perception [6].

1.1 Challenges in Face Recognition

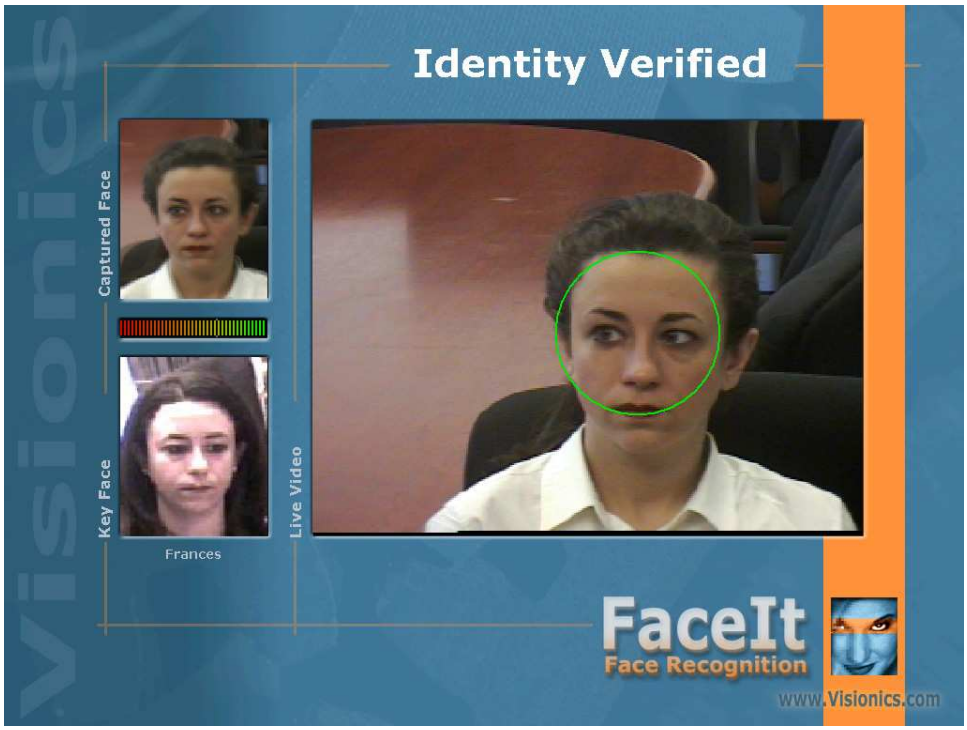
Humans can easily recognize a known face in various conditions and representations (see Fig. 1.3). Such a remarkable ability of humans to recognize faces with large intra-subject variations has inspired vision researchers to develop automated systems for face recognition based on 2D face images. However, the current state-of-the-art machine vision systems can recognize faces only in a constrained environment. Note that there are two types of face comparison scenarios, called (i) face *verification (or authentication)* and (ii) face *identification (or recognition)*. As shown in Fig. 1.4, face verification involves a one-to-one match that compares a query face image against a template face image whose identity is being claimed, while face identification involves one-to-many matches that compare a query face image against all the template images in a face database to determine the identity of the query face. The main chal-



(a)

(b)

Figure 1.1. Applications using face recognition technology: (a) and (b) automated video surveillance (downloaded from Visionics [1] and FaceSnap [2], respectively); (c) and (d) access control (from Visionics [1] and from Viisage [3], respectively); (e) management of photo databases (from Viisage [3]); (f) multimedia communication (from Eyematic [4]). *Images in this dissertation are presented in color.*



(c)

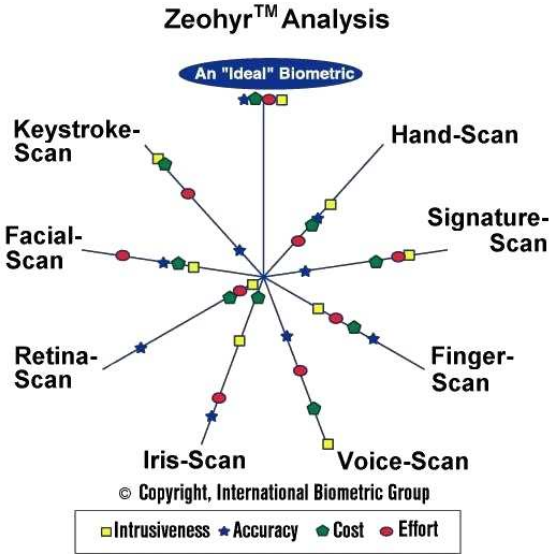
(d)

Figure 1.1. (Cont'd).

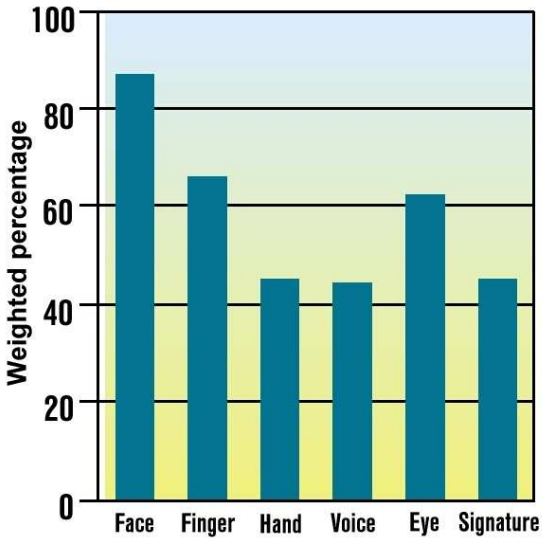
(e)

(f)

Figure 1.1. (Cont'd).



(a)



(b)

Figure 1.2. Comparison of various biometric features: (a) based on zephyr analysis (downloaded from [5]); (b) based on MRTD compatibility (from [6]).

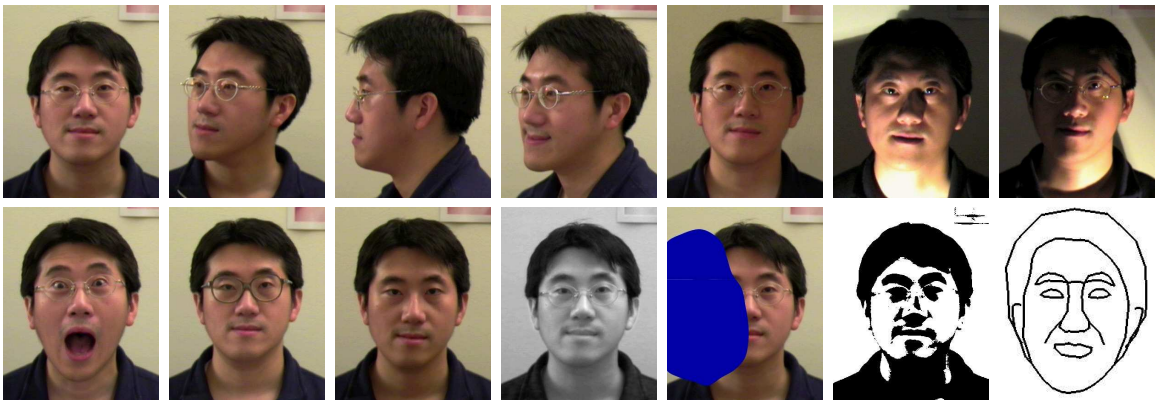
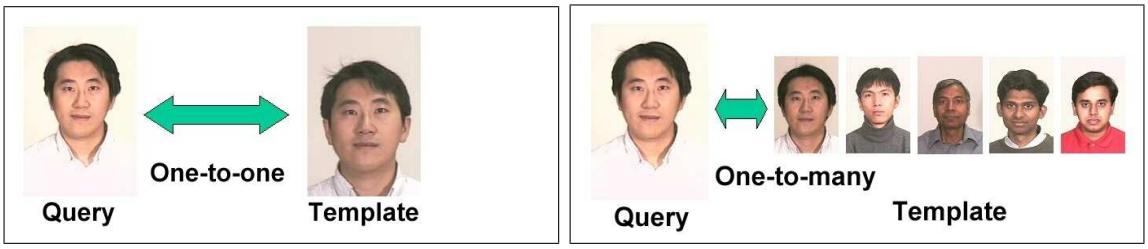


Figure 1.3. Intra-subject variations in pose, illumination, expression, occlusion, accessories (e.g., glasses), color, and brightness.



(a) (b)

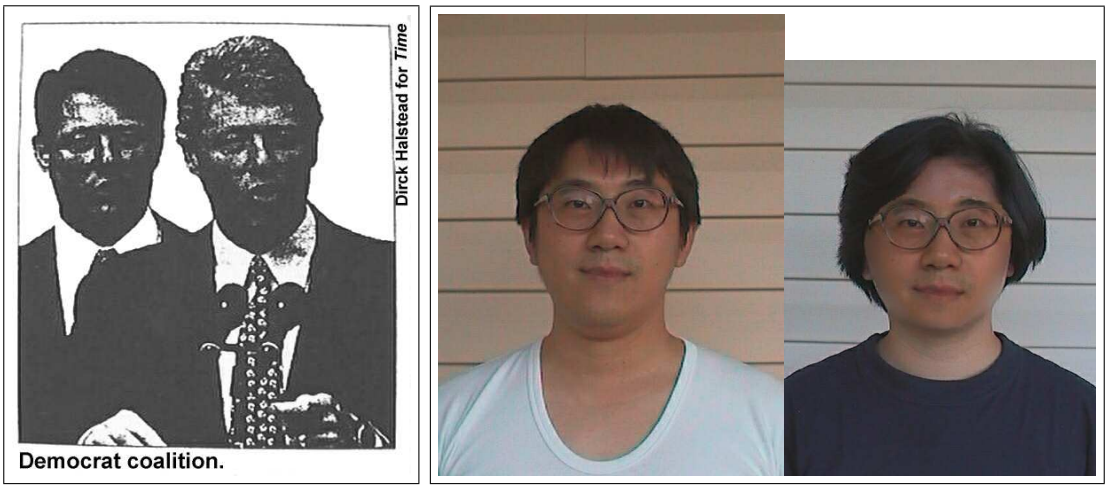
Figure 1.4. Face comparison: (a) face verification/authentication; (b) face identification/recognition. Face images are taken from the MSU face database [7].

lenge in vision-based face recognition is the presence of a high degree of variability in human face images. There can be potentially very large intra-subject variations (due to 3D head pose, lighting, facial expression, facial hair, and aging [41]) and rather small inter-subject variations (due to the similarity of individual appearances). Currently available vision-based recognition techniques can be mainly categorized into two groups based on the face representation which they use: (i) appearance-based which use holistic texture features, and (ii) geometry-based which use geometrical features of the face. Experimental results show that appearance-based methods generally perform a better recognition task than those based on geometry, because it is difficult to robustly extract geometrical features especially in face images of low

resolutions and of poor quality (i.e., to extract features under uncertainty). However, the appearance-based recognition techniques have their own limitations in recognizing human faces in images with wide variations in 3D head pose and in illumination [38]. Hence, in order to overcome variations in pose, a large number of face recognition techniques have been developed to take into account the 3D face shape, extracted either from a video sequence or range data. As for overcoming the variations in illumination, several studies have explored features such as edge maps (e.g., eigenhills and eigenedges in [42]), intensity derivatives, Gabor-filter responses [43], and the orientation fields of intensity gradient [44]. However, none of these approaches by themselves lead to satisfactory recognition results. Hence, the explicit 3D face model combined with its reflectance model is believed to be the best representation of human faces for the appearance-based approach [43].

1.2 Semantic Facial Components

Face recognition technology provides useful tools for content-based image and video retrieval based on a semantic (high-level) concept, i.e., human faces. Is all face processing holistic [45]? Some approaches, including feature-based and appearance-based [46] methods, emphasize that internal facial features (i.e., pure face regions) play the most important role in face recognition. On the other hand, some appearance-based methods suggest that **in some situations** face recognition is better interpreted as *head recognition* [8], [31]. An example supporting the above argument was demonstrated for Clinton and Gore heads [8] (See Fig. 1.5(a)). While the two faces in



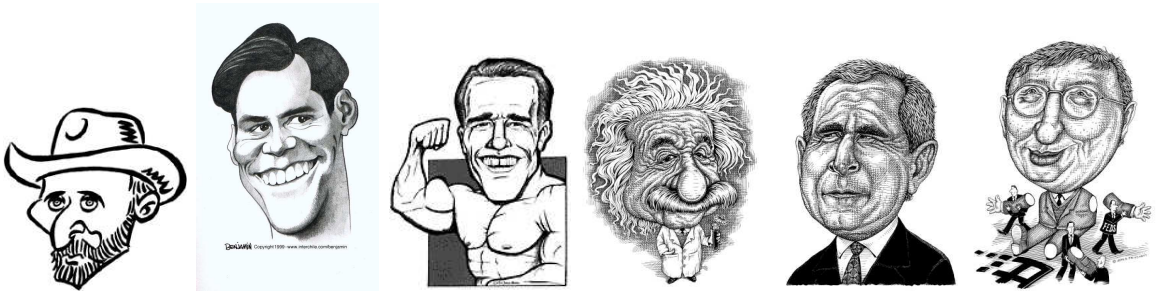
(a) (b)

Figure 1.5. Head recognition versus face recognition: (a) Clinton and Gore heads with the same internal facial features, adapted from [8]; (b) two faces of different subjects with the same internal facial components show the important role of hair and face outlines in human face recognition.

Fig. 1.5(a) have identical internal features, we can still distinguish Clinton from Gore. We notice that in this "example" *the hair style and the face outline are significantly different*. We reproduce this scenario, across genders, in Fig. 1.5(b). Humans will usually identify these two persons with different identities. This prompted Liu et al. [47] to emphasize that there is no use of face masks (to remove the "non-pure-face" portion) in their appearance-based method. As a result, we believe that the separation of external and internal facial features/components is helpful in assigning weights on external and internal facial features in the face recognition process.

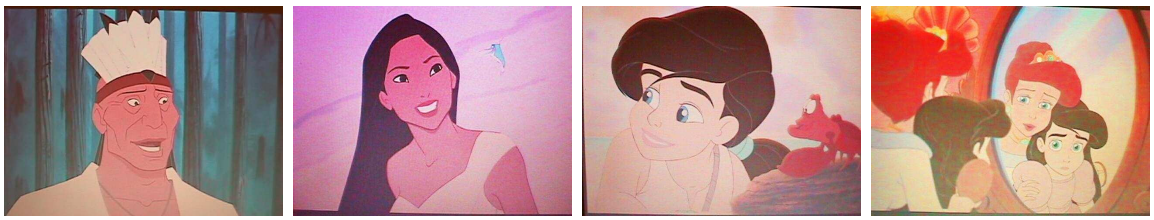
Modeling facial components at the semantic level (i.e., eyebrows, eyes, nose, mouth, face outline, ears, and the hair outline) helps to separate external and internal facial components, and to understand how these individual components contribute to face recognition. Examples of modeling facial components can be found in the faces represented in caricatures and cartoons. However, the fact that humans

can recognize known faces in caricature drawings (e.g., faces shown in Fig. 1.6) and cartoons (see Fig.1.7) without any difficulty has not been fully explored in research studies on face recognition [48], [49], [50], [51]. Note that some of the faces shown



(a) (b) (c) (d) (e) (f)

Figure 1.6. Caricatures of (a) Vincent Van Gogh; (b) Jim Carrey; (c) Arnold Schwarzenegger; (d) Einstein; (e) G. W. Bush; and (f) Bill Gates. Images are downloaded from [9], [10] and [10]. Caricatures reveal the use of component weights in face identification.



(a) (b) (c) (d)

Figure 1.7. Cartoons reveal that humans can easily recognize characters whose facial components are depicted by simple line strokes and color characteristics: (a) and (b) are frames adapted from the movie Pocahontas; (c) and (d) are frames extracted from the movie Little Mermaid II. (Disney Enterprises, Inc.)

in Fig. 1.6 are represented only by strokes (geometrical features), while some others have parts of facial features dramatically emphasized with some distortion. Cartoon faces are depicted by line drawings and color without shading. People can easily identify faces in caricatures (see, Fig. 1.6) that exaggerate some of the facial components/landmarks. Besides, we can also easily identify known faces merely based on some salient facial components. For example, we can quickly recognize a known face

with a distinctive chin no matter whether the face appears in a caricature (e.g., Jim Carrey shown in Fig. 1.6(b)) or in a real photo [52]. Caricatures reveal that there are certain facial features which are salient for each individual and that a relatively easier identification of faces can occur by emphasizing distinctive facial components (using weights) and their configuration. Besides, the spatial configuration of facial components has been shown to take a more important role in face recognition than local texture by using inverted faces [53] in which the (upright) face recognition is disrupted (see Fig. 1.8). Therefore, we group these salient facial components [48] as

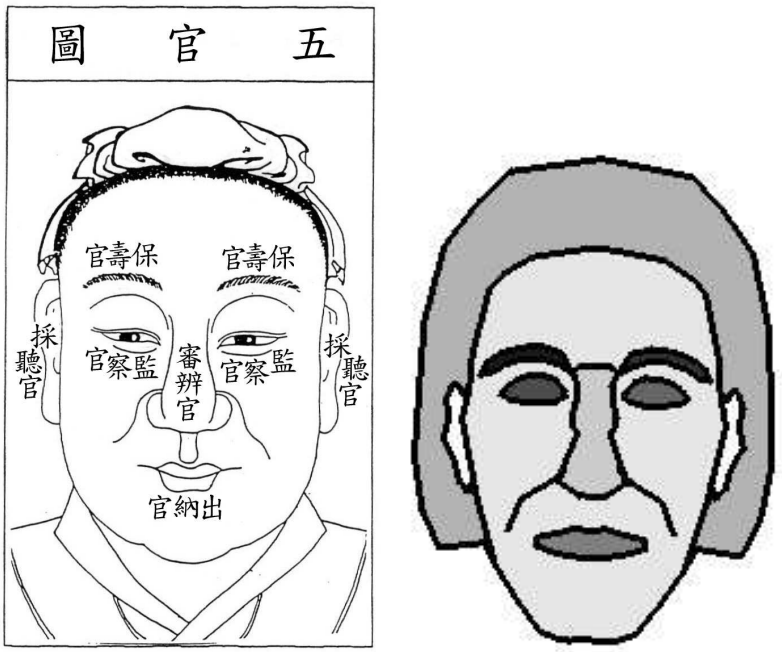
(a) (b) (c)

Figure 1.8. Configuration of facial components: (a) face image; (b) face image in (a) with enlarged eyebrow-to-eye and nose-to-mouth distances; (c) inverted face of the image in (b). A small change of component configuration results in a significantly different facial appearance in an upright face in (b); however, this change may not be perceived in an inverted face in (c).

a graph and derive *component weights* in our face matching algorithm to improve the recognition performance.

In addition, humans can recognize faces in the presence of occlusions, i.e., face recognition can be based on a (selected) subset of facial components. This explains

the motivation for studies that attempt to recognize faces from eyes only [54]. The use of component weights can facilitate face recognition based on selected facial components. Furthermore, the shape of facial components (see Fig. 1.9(a)) has been used in physiognomy (or face reading, an ancient art of deciphering a person's past and personality from his/her face). In light of this art, we design a semantic face graph for face recognition (see in Chapter 5), shown in Fig. 1.9(b), in which ten facial components are filled with different shades in a frontal view.



(a) (b)

Figure 1.9. Facial features/components: (a) five kinds of facial features (i.e., eyebrows, eyes, nose, ears, and mouth) in a face for reading faces in physiognomy (downloaded from [11]); (b) a frontal semantic face graph, whose nodes are facial components that are filled with different shades.

For each facial component, the issue of representation also plays an important role in face recognition. It has been believed that local facial texture and shading are crucial for recognition [52]. However, some frames of a cartoon video, as shown in Fig.

1.7, reveal that line drawings and color characteristics (shades) of facial components (e.g., dark colors for eyebrows and both bright and dark colors for eyes) provide sufficient information for humans to recognize the faces of characters in cartoons. People can even recognize cartoon faces without the use of shading information, which is rather unstable under different lighting conditions. Consequently, we believe that curves (or sketches) and shades of facial components provide a promising solution to the representation of facial components for recognition. However, very little work has been done in face recognition based on facial sketches [55], [56] and (computer-generated [57]) caricatures [58], [48], [50].

In summary, external and internal facial components, and distinctiveness, configuration and local texture of facial components all contribute to the process of face recognition. Humans can *seamlessly blend* and *independently perform* appearance-based and geometry-based recognition approaches efficiently. Therefore, we believe that merging [59], [60] the holistic texture features and the geometrical features (especially at a semantic level) is a promising method to represent faces for recognition. While we focus on the 3D variations in faces, we should also take the temporal (aging) factor into consideration while designing face recognition systems [41]. In addition to large intra-subject variations, another difficulty in recognizing faces lies in the small inter-subject variations (shown in Fig. 1.10). Different persons may have very similar appearances. Identifying people with very similar appearances remains a challenging task in automatic face recognition.

(a) (b)

Figure 1.10. Similarity of frontal faces between (a) twins (downloaded from [12]); and (b) a father and his son (downloaded from [13]).

1.3 Face Recognition Systems

Face recognition applications in fact involve several important steps, such as face detection for locating human faces, face tracking for following moving subjects, face modeling for representing human faces, face coding/compression for efficiently archiving and transmitting faces, and face matching for comparing represented faces and identifying a query subject. Face detection is usually an important first step. Detecting faces can be viewed as a two-class (face vs. non-face) classification problem, while recognizing faces can be regarded as a multiple-class (multiple subjects) classification problem within the face class. Face detection involves certain aspects of face recognition mechanism, while face recognition employs the results of face detection. We can consider face detection and recognition as the first and the second stages in a sequential classification system. The crucial issue here is to determine an appropriate feature space to represent a human face in such a classification system. We believe that a seamless combination of face detection, face modeling, and recognition algorithms has the potential of achieving high performance for face identification

applications.

With this principle, we propose two automated recognition paradigms, shown in Fig. 1.11 and Fig. 1.12, that can combine face detection as well as *tracking* (not included in this thesis, but can be realized based on our current work), modeling, and recognition. The first paradigm requires both video sequences and 2.5D/3D facial measurements as its input in the learning/enrollment stage. In the recognition/test stage, however, face images are extracted from video input only. Faces are identified based on an appearance-based algorithm. The second paradigm requires only video sequences as its input in both learning and recognition stages. Its face recognition module makes use of a semantic face matching algorithm to compare faces based on weighted facial components.

Both paradigms contain three major modules: (i) face detection and feature extraction, (ii) face modeling, and (iii) face recognition. The face detection/location and feature extraction module is able to locate faces in video sequences. The most important portion of this module is a feature extraction sub-module that extracts geometrical features (such as face boundary, eyes, eyebrows, nose, and mouth), and texture/color features (estimation of the head pose and illumination is left as a future research direction). The face modeling module employs these extracted features for modifying the generic 3D face model in the learning and recognition stages. In this thesis, we describe the implementation of the face modeling module in both proposed paradigms for the frontal view only. The extension of the face modeling module to non-frontal views can be a future research direction. The recognition module makes use of facial features extracted from an input image and the learned 3D models to

verify the face present in an image in the recognition stage. This thesis has developed a robust face detection module which is used to facilitate applications such as face tracking for surveillance, and face modeling for identification (as well as verification). We will briefly discuss the topics of face detection and recognition, face modeling as well as compression, and face-based image retrieval in the following sections.

1.4 Face Detection and Recognition

Human activity is a major concern in a wide variety of applications such as video surveillance, human computer interface, face recognition [37], [36], [38], and face image database management [40]. Detecting faces is a crucial step and usually the first one in these identification applications. However, due to various head poses, illumination conditions, occlusion, and distances between teh sensor and the subject (which may result in a blurred face), detecting human faces is an extremely difficult task under unconstrained environments (see images in Figs. 1.13 (a) and (b)). Most face recognition algorithms assume that the problem of face detection has been solved, that is, the face location is known. Similarly, face tracking algorithms (e.g., [61]) often assume the initial face location is known. Since face detection can be viewed as a two-class (face vs. non-face) classification problem, some techniques developed for face recognition (e.g., holistic/template approaches [21], [62], [63], [64], feature-based approaches [65], and their combination [66]) have been used to detect faces. However, these detection techniques are computationally very demanding and cannot handle large variations in faces. In addition to the face location, a face detection

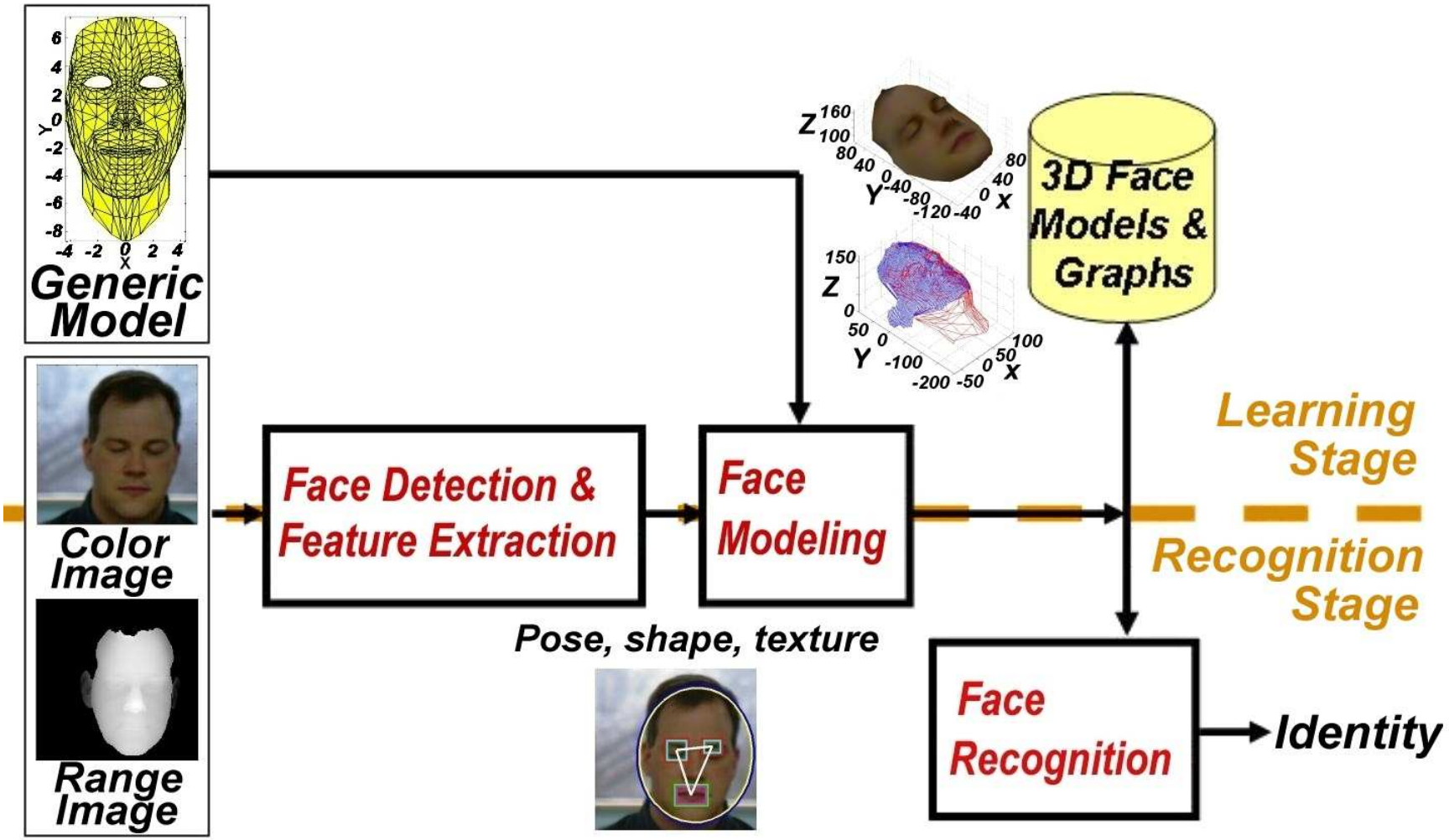


Figure 1.11. System diagram of our 3D model-based face recognition system using registered range and color images.

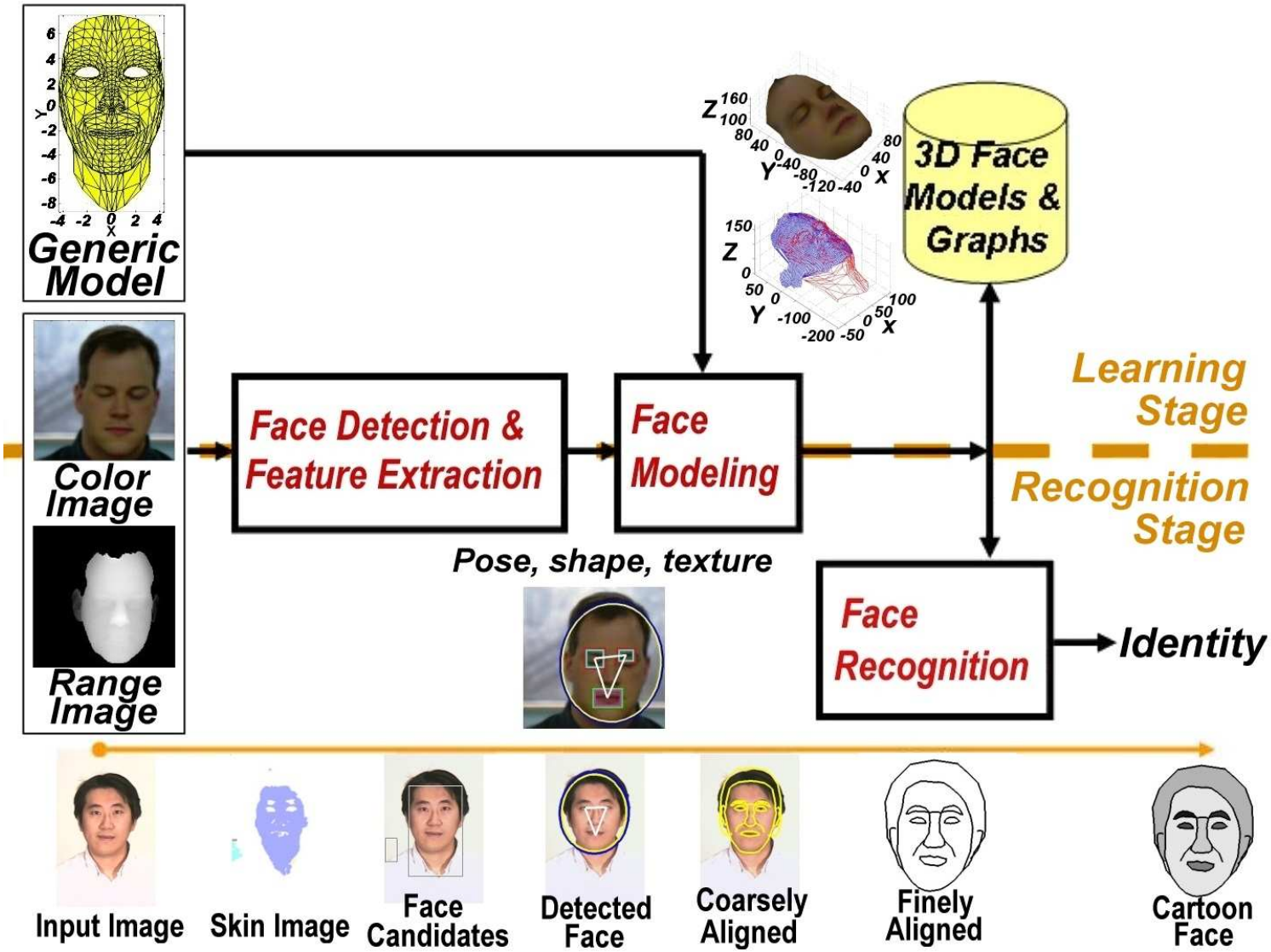


Figure 1.12. System diagram of our 3D model-based face recognition system without the use of range data.

(a)

(b)

Figure 1.13. Face images taken under unconstrained environments: (a) a crowd of people (downloaded from [14]); (b) a photo taken at a swimming pool.

algorithm can also provide geometrical facial features for face recognition. Merging the geometrical features and holistic texture (appearance-based) features is believed to be a promising method of representing faces for recognition [59], [60]. Therefore, we believe that a seamless combination of face detection and recognition algorithms has the potential of providing a high performance face identification algorithm.

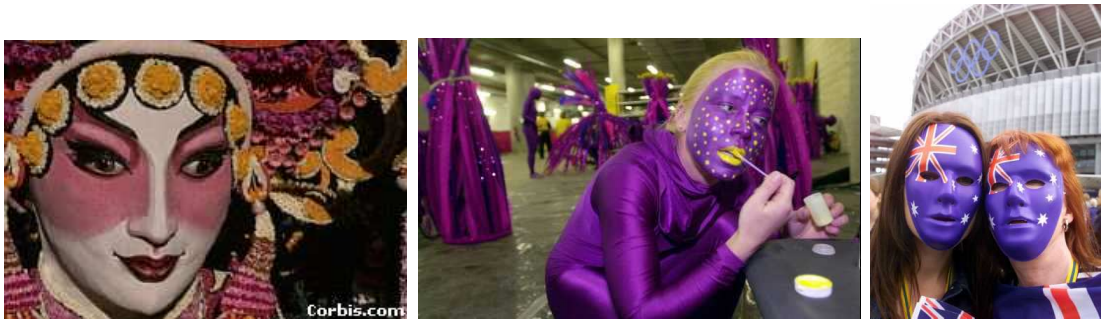
Hence, we have proposed a face detection algorithm for color images, which is able to generate geometrical as well as texture features for recognition. Our approach is based on modeling skin color and extracting geometrical facial features. The skin color is detected by using a lighting compensation technique and a nonlinear color transformation. The geometrical facial features are extracted from eye, mouth, and face boundary maps. The detected faces, including the extracted facial features, are organized as a graph for modeling and recognition processes. Our algorithm can detect faces under different head poses, illuminations, and expressions (see Fig. 1.14(a)), and family photos (see Fig. 1.14(b)). However, our detection algorithm is not designed for detecting faces in gray-scale images, cropped face images (see Fig. 1.15(a)) and faces wearing make-up or mask (see Figs. 1.15(b) and (c)).

1.5 Face Modeling for Recognition

Our face recognition systems are based on 3D face models. 3D models of human faces have been widely used to facilitate applications such as video compression/coding, human face tracking, facial animation, augmented reality, recognition of facial expression, and face recognition. Figure 1.16 shows two graphical user interfaces of a

(a) (b)

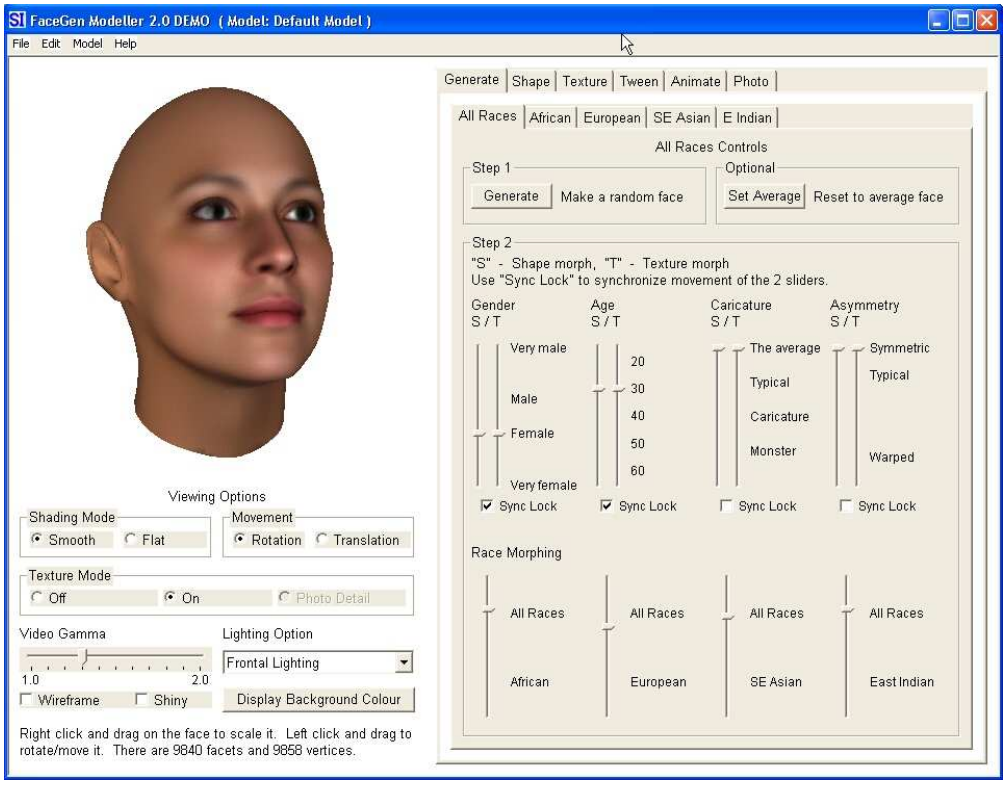
Figure 1.14. Face images for our detection algorithm: (a) a montage image containing images adapted from MPEG7 content set [15]; (b) a family photo.



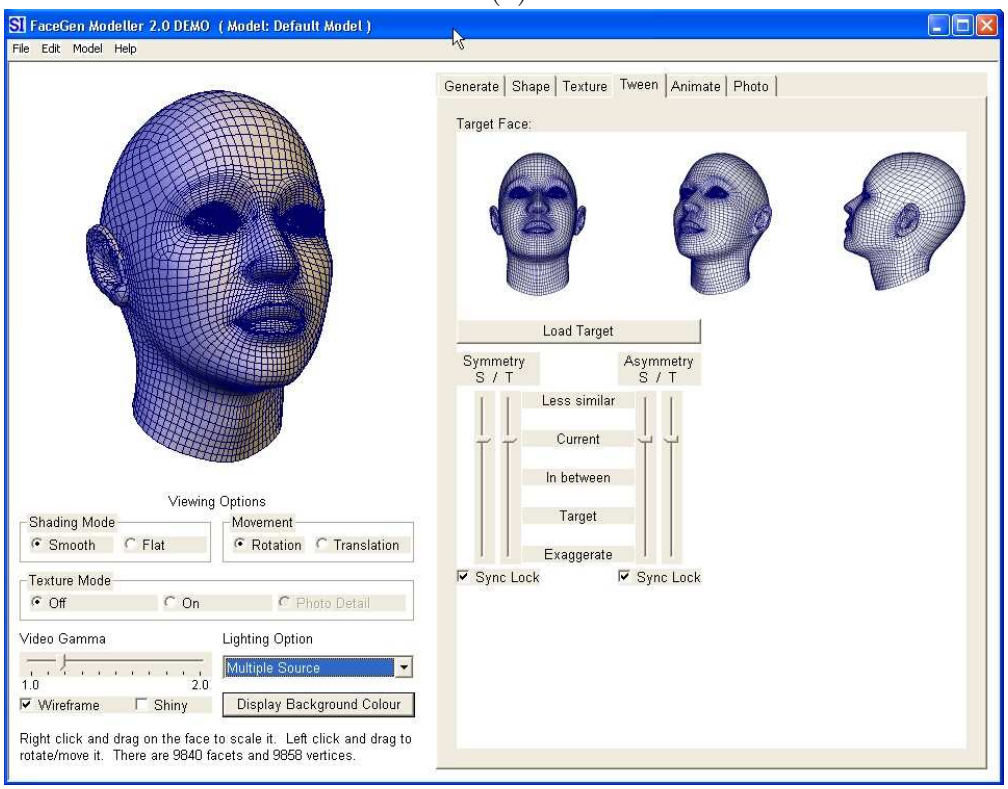
(a) (b) (c)

Figure 1.15. Face images not suitable for our detection algorithm: (a) cropped image (downloaded from [16]); (b) a performer wearing make-up (from [14]); (c) people wearing face masks (from [14]).

commercial parametric face modeling system [17], FaceGen Modeller, which is based on face shape statistics. It can efficiently create a character with specified age, gender, race, and caricature morphing. Current trend in face recognition is to employ 3D face model explicitly [67], because such a model provides a potential solution to identifying faces with variations in illumination, 3D head pose, and facial expression. These variations, called the intra-subject variations, also include changes due to aging, facial hair, cosmetics, and facial accessories. These intra-subject variations constitute the primary challenges in the field of face recognition. As object-centered representations of human faces, 3D face models not only can augment recognition systems that utilize viewer-centered face representations (based on 2D face images), but also can blend together holistic approaches and geometry-based approaches for recognition. However, the three state-of-the-art face recognition algorithms [68], (1) the principal component analysis (PCA)-based algorithm; (2) the local feature analysis (LFA)-based algorithm; and (3) the dynamic-link-architecture-based algorithm, use only viewer-centered representations of human faces. A 3D model-based matching algorithm is likely to provide a potential solution for advancing face recognition technology. However, for face recognition, it is more important to capture facial distinctiveness of recognition-oriented components than to generate a realistic face model. We briefly introduce our face modeling methods for recognition (i.e., face alignment) and model compression in the following subsections.



(a)



(b)

Figure 1.16. Graphical user interfaces of the FaceGen modeller [17]. A 3D face model shown (a) with texture mapping; (b) with wireframe overlaid.

1.5.1 Face Alignment Using 2.5D Snakes

In our first recognition system (shown in Fig. 1.11), we have proposed a face modeling method which adapts an existing generic face model (a priori knowledge of a human face) to an individual's facial measurements (i.e., range and color data). We use the face model that was created for facial animation by Waters [69] as our generic face model. Waters' model includes details of facial features that are crucial for face recognition. Our modeling process aligns the generic model onto extracted facial features (regions), such as eyes, mouth, and face boundary, in a global-to-local way, so that facial components that are crucial for recognition are fitted to the individual's facial geometry. Our global alignment is based on the detected locations of facial components, while the local alignment utilizes two new techniques which we have developed, *displacement propagation* and *2.5D active contours*, to refine local facial components and to smoothen the face model. Our goal of face modeling is to generate a learned 3D model of an individual for verifying the presence of the individual in a face database or in a video. The identification process involves (i) the modification of the learned 3D model based on different head poses and illumination conditions and (ii) the matching between 2D projections of the modified 3D model, whose facial shape is integrated with facial texture, and sensed 2D facial appearance.

1.5.2 Model Compression

Requirements of easy manipulation, progressive transmission, effective visualization and economical storage for 3D (face) models have resulted in the need for *model*

compression. The complexity of an object model depends not only on object geometry but also on the choice of its representation. The 3D object models explored in computer vision and graphics research have gradually evolved from simple polyhedra, generated in mechanical Computer Aided Design (CAD) systems, to complex free-form objects, such as human faces captured from laser scanning systems. Although human faces have a complex shape, modeling them is useful for emerging applications such as virtual museums and multimedia guidebooks for education [70], [71], low-bandwidth transmission of human face images for teleconferencing and interactive TV systems [72], virtual people used in entertainment [73], sale of facial accessories in e-commerce, remote medical diagnosis, and robotics and automation [74].

The major reason for us to adopt the triangular mesh as our generic human face model is that it is suitable for describing and simplifying the complexity of facial geometry. In addition, there are a number of geometry compression methods available for compressing triangular meshes (e.g., the topological surgery [75] and the multiresolution mesh simplification [76]). Beside these helps, we can further obtain a more compact representation of a 3D face model by carefully selecting vertices of the triangular mesh for representing facial features that are extracted for face recognition. Our proposed semantic face graph used in the semantic recognition paradigm (see Fig. 1.12) is such an example.

1.5.3 Face Alignment Using Interacting Snakes

For the semantic recognition system (shown in Fig. 1.12), we define a *semantic face graph*. A semantic face graph is derived from a generic 3D face model for identifying faces at the semantic level. The nodes of a semantic graph represent high-level facial components (e.g., eyes and mouth), whose boundaries are described by open (or closed) active contours (or snakes). In our recognition system, face alignment plays a crucial role in adapting a priori knowledge of facial topology, encoded in semantic face graph, onto the sensed facial measurements (e.g., face images). The semantic face graph is first projected onto a 2D image, coarsely aligned to the output of the face detection module, and then finely adapted to the face images using interacting snakes.

Snakes are useful models for extracting the shape of deformable objects [77]. Hence, we model the component boundaries of a 2D semantic face graph as a collection of snakes. We propose an approach for manipulating multiple snakes iteratively, called *interacting snakes*, that minimizes the attraction energy functionals on both contours and enclosed regions of individual snakes and the repulsion energy functionals among multiple snakes that interact with each other. We evaluate the interacting snakes through two types of implementations, explicit (parametric active contours) and implicit (geodesic active contours) curve representations, for face alignment.

Once the semantic face graph has been aligned to face images, we can derive component weights based on distinctiveness and visibility of individual components. The aligned face graph can also be easily used to generate cartoon faces and facial

caricatures by exaggerating the distinctiveness of facial components. After alignment, facial components are transformed to a feature space spanned by Fourier descriptors of facial components for face recognition, called *semantic face matching*. The matching algorithm computes the similarity between semantic face graphs of face templates in a database and a semantic face graph that is adapted to a given face image. The semantic face graph allows face matching based on selected facial components, and effective 3D model updating based on 2D face images. The results of our face matching demonstrate that the proposed face model can lead to classification and visualization (e.g., the generation of cartoon faces and facial caricatures) of human faces using the derived semantic face graphs.

1.6 Face Retrieval

Today, people can accumulate a large number of images and video clips (digital content) because of the growing popularity of digital imaging devices, and because of the decreasing cost of high-capacity digital storage. This significant increase in the amount of digital content requires database management tools that allow people to easily archive and retrieve contents from their digital collections. Since humans and their activities are typically the subject of interest in consumers' images and videos, detecting people and identifying them will help to automate image and video archival based on a high-level semantic concept, i.e., human faces. For example, we can design a system that manages digital content of personal photos and amateur videos based on the concept of human faces, e.g., "retrieve all images containing Carrie's face."

Using merely low-level features (e.g., skin color or color histograms) for retrieval and browsing is neither robust nor acceptable to the user. High level semantics have to be used to make such an image/video management system useful. Fig. 1.17 shows a graphical user interface of a facial feature-based retrieval system [18].

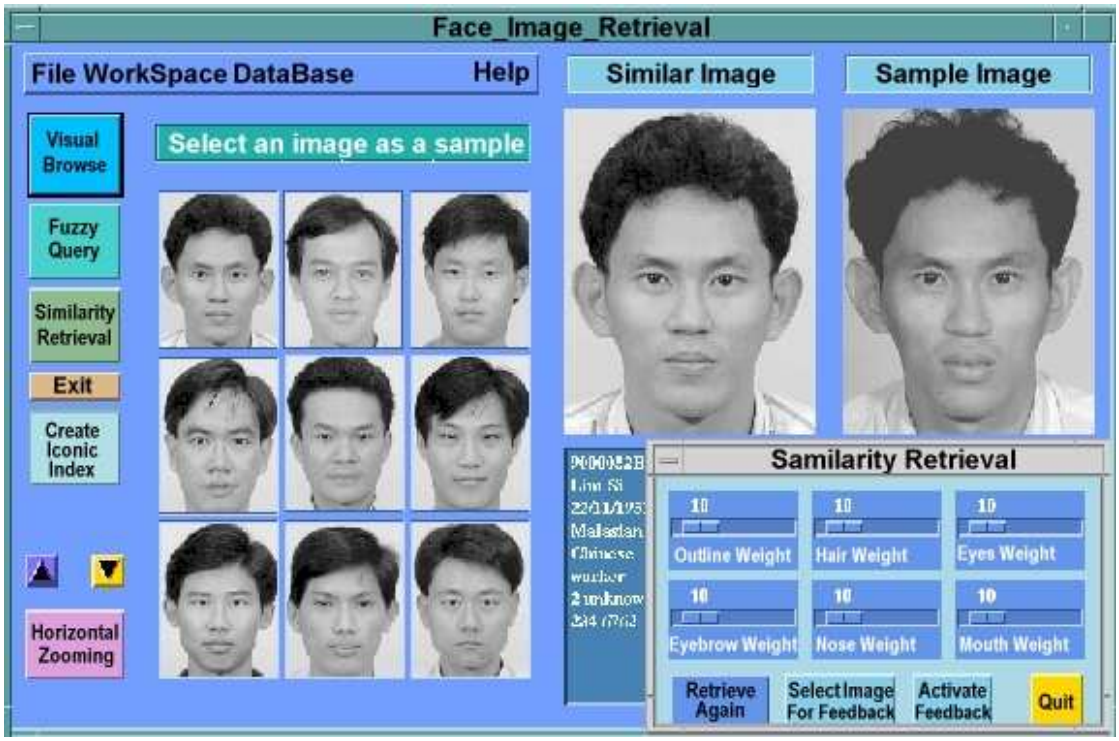


Figure 1.17. A face retrieval interface of the FACEit system [18]: the system gives the most similar face in a database given a query face image.

In summary, the ability to group low-level features as a meaningful semantic entity is a critical issue in the retrieval of visual content. Accurately and efficiently detecting human faces plays a crucial role in facilitating face identification for managing face databases. In face recognition algorithms, the high-level concept–a human face–is implicitly expressed by face representations such as locations of feature points, surface texture, 2D graphs with feature nodes, 3D head surface, and combinations of them. The face representation plays an important role in the recognition process because

different representations lead to different matching algorithms. We can design a database management system that utilizes the outputs of our face detection and modeling modules as indices to search a database based on the semantic concepts, such as "find all the images containing John's faces" and "search faces which have Vincent's eyes (or face shape).

1.7 Outline of Dissertation

This dissertation is organized as follows. Chapter 2 presents a brief literature review on face detection and recognition, face modeling (including model compression), and face retrieval. In Chapter 3, we present our face detection algorithm for color images. Chapter 4 discusses our range data-based face modeling method for recognition. Chapter 5 describes the semantic face recognition system, including face alignment using interacting snakes, a semantic face matching algorithm, and the generation of cartoon faces and facial caricatures. Chapter 6 presents conclusions and future directions related to this work.

1.8 Dissertation Contributions

The major contributions of this dissertation are categorized into the topics of face detection, face modeling, and face recognition. In **face detection**, we have developed a new face detection algorithm for multiple non-profile-view faces with complex background in color images, based on localization of skin-tone color and facial features

such as eyes, mouth and face boundary. The main properties of this algorithms are listed as follows.

- **Lighting compensation**: This method corrects the color bias and recovers the skin-tone color by automatically estimating the reference white pixels in a color image, under the assumption that an image usually contains "real white" (i.e., white reference) pixels and the dominant bias color in an image always appears as "real white".

- **Non-linear color transformation**: In literature, the chrominance components of the skin tone have been assumed to be independent of the luminance component of the skin tone. We found that the chroma of skin tone depends on the luma. We overcome the difficulty of detecting the low-luma and high-luma skin tone colors by applying a nonlinear transform to the YC_bC_r color space. The transformation is based on the linearly fitted boundaries of our training skin cluster in YC_b and YC_r color subspaces.

- **Modeling a skin-tone color classifier as an elliptical region**: A simple classifier which constructs an elliptical decision region in the chroma subspace, C_bC_r, has been designed, under the assumption of the Gaussian distribution of skin tone color.

- **Construction of facial feature maps for eyes, mouth, and face boundary**: With the use of gray-scale morphological operators (dilation and erosion), we construct these feature maps by integrating the luminance and chrominance information of facial features. For example, eye regions have high C_b (difference

between blue and green colors) and low C_r (difference between red and green colors) values in chrominance components, and have brighter and darker values in the luminance component.

- **Construction of a diverse database of color images for face detection**: The database includes a MPEG7 content set, mug-shot style web photos, family photos, and news photos.

In **face modeling**, we have designed two methods for aligning a 3D generic face model onto facial measurements captured in the frontal view: one uses facial measurements of registered color and range data; the other merely uses color images. In the first method, we have developed two techniques for face alignment:

- **2.5D snake**: A 2.5D snake is designed to locally adapt a contour to each facial component. The design of snake includes an iterative deformation formula, placement of initial contours, and the minimization of energy functional. We reformulated 2D active contours (a dynamic programming approach) for 3D contours of eye, nose, mouth, and face boundary regions. We have constructed initial contours based on the outputs of face detection (i.e., locations of the face and facial components). We form energy maps for individual facial components based on 2D color image and 2.5D range data, hence the name 2.5D snake.

- **Displacement propagation**: This technique is designed to propagate the displacement of a group of vertices on a 3D face model from contour points on facial components to other points on non-facial components. The propagation

can be applied to a 3D face model whenever a facial component is coarsely relocated or is finely deformed by the 2.5D snake.

In the second face modeling method, we developed a technique for face alignment:

- **Interacting snakes**: The snake deformation is formulated by a finite difference approach. The initial snakes for facial components are obtained from the 2D projection of the semantic face graph on a generic 3D face model. We have designed the *interacting snakes* technique for manipulating multiple snakes iteratively that minimizes the attraction energy functionals on both contours and enclosed regions of individual snakes and minimizes the repulsion energy functionals among multiple snakes.

In **face recognition**, we have proposed two paradigms as shown in Figs. 1.11 and 1.12.

- **The first (range data-based) recognition paradigm**: This paradigm is designed to *automate* and *augment* appearance-based face recognition approaches based on 3D face models. In this system, we have integrated our face detection algorithm, face modeling method using the 2.5D snake, and an appearance-based recognition method using the hierarchical discriminant regression [78]. However, the recognition module can be replaced with other appearance-based algorithms such as PCA-based and LDA-based methods. The system can learn a 3D face model for an individual, and generate an arbitrary number of 2D face images under different head poses and illuminations (can be extended to different expressions) for training an appearance-based face classifier.

- **The second (semantic) recognition paradigm**: This paradigm is designed to automate the face recognition process at *a semantic level* based on the distinctiveness and visibility of facial components in a given face image captured in near frontal views. (This paradigm can be extended to face images taken in non-frontal views). We have decomposed a generic 3D face model into recognition-oriented facial components and non-facial components, and formed a 3D semantic face graph for representing facial topology and extracting facial components. In this recognition system, we have integrated our face detection algorithm, our face modeling method using interacting snakes, and our semantic face matching algorithm. The recognition can be achieved at a semantic level (e.g., comparing faces based on eyes and the face boundary only) due to the alignment of facial components. We have also introduced component weights, which play a crucial role in face matching, to emphasize component's distinctiveness and visibility. The system can generate cartoon faces from aligned semantic face graphs and facial caricatures based on an averaged face graph for face visualization.

Chapter 2

Literature Review

We first review the development of face detection and recognition approaches, followed by a review of face modeling and model compression methods. Finally, we will present one major application of face recognition technology, namely, face retrieval. We primarily focus on the methods that employ the task-specific cognition or behaviors specified by humans (i.e., artificial intelligence pursuits), although there are developmental approaches for facial processing (e.g., autonomous mental development [79] and incremental learning [80] methods) that have emerged recently.

2.1 Face Detection

Various approaches to face detection are discussed in [19], [20], [81],[82], and [83]. The major approaches are listed chronologically in Table 2.1 for a comparison. For recent surveys on face detection, see [82] and [83]. These approaches utilize techniques such as principal component analysis (PCA), neural networks, machine learning, infor-

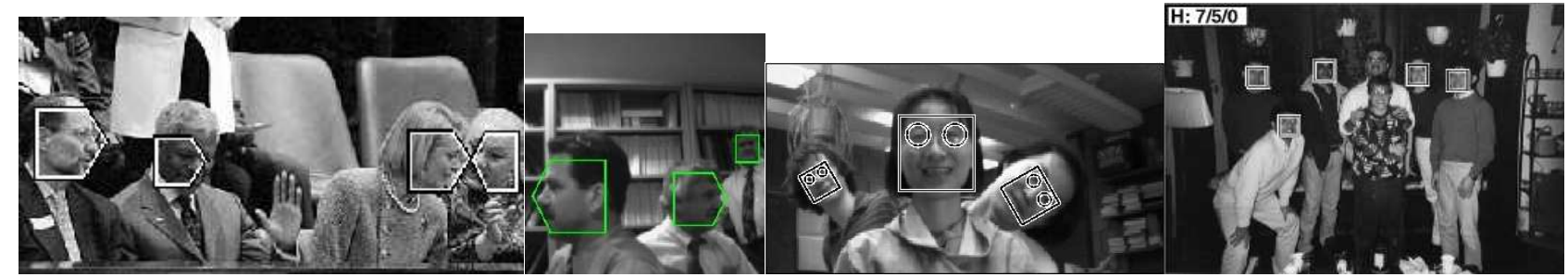
Table 2.1
Summary of various face detection approaches.

Authors	Year	Approach	Features Used	Head Pose	Test Databases	Minimal Face Size
Féraud et al. [19]	2001	Neural networks	Motion; Color; Texture	Frontal to profile	Sussex; CMU; Web images	15×20
DeCarlo et al. [61]	2000	Optical flow	Motion; Edge; Deformable face model; Texture	Frontal to profile	Videos	NA
Maio et al. [20]	2000	Facial templates; Hough transform	Texture; Directional images	Frontal	Video images	20×27
Abdel-Mottaleb et al. [84]	1999	Skin model; Feature	Color	Frontal to profile	HHI	13×13
Garcia et al. [21]	1999	Statistical wavelet analysis	Color; Wavelet coefficients	Frontal to near frontal	MPEG videos	80×48
Wu et al. [85]	1999	Fuzzy color models; Template matching	Color	Frontal to profile	Still color images	20×24
Rowley et al. [24], [23]	1998	Neural networks	Texture	(Upright) frontal	FERET; CMU; Web images	20×20
Sung et al. [25]	1998	Learning	Texture	Frontal	Video images; newspaper scans	19×19
Colmenarez et al. [86]	1997	Learning	Markov processes	Frontal	FERET	11×11
Yow et al. [26]	1997	Feature; Belief networks	Geometrical facial features	Frontal to profile	CMU	60×60
Lew et al. [27]	1996	Markov random field; DFFS [64]	Most informative pixel	Frontal	MIT; CMU; Leiden	23×32

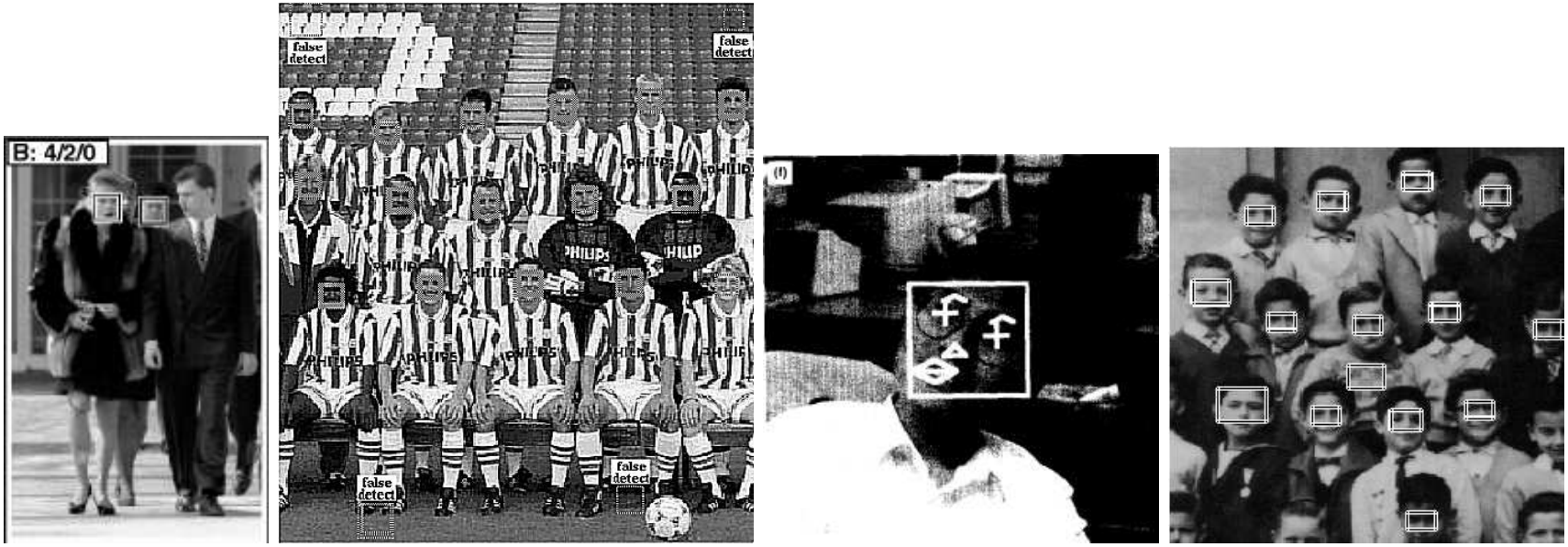
mation theory, geometrical modeling, (deformable) template matching, Hough transform, extraction of geometrical facial features, motion extraction, and color analysis. Typical detection outputs are shown in Fig. 2.1. In these images, a detected face is usually overlaid with graphical objects such as a rectangle or an ellipse for a face, and circles or crosses for eyes. The neural network-based [24], [23] and the view-based [25] approaches require a large number of face and non-face training examples, and are designed primarily to locate frontal faces in grayscale images. It is difficult to enumerate "non-face" examples for inclusion in the training databases. Schneiderman and Kanade [22] extend their learning-based approach for the detection of frontal faces to profile views. A feature-based approach combining geometrical facial features with belief networks [26] provides face detection for non-frontal views. Geometrical facial templates and the Hough transform were incorporated to detect grayscale frontal faces in real time applications [20]. Face detectors based on Markov random fields [27], [87] and Markov chains [88] make use of the spatial arrangement of pixel gray values. Model based approaches are widely used in tracking faces and often assume that the initial location of a face is known. For example, assuming that several facial features are located in the first frame of a video sequence, a 3D deformable face model was used to track human faces [61]. Motion and color are very useful cues for reducing search space in face detection algorithms. Motion information is usually combined with other information (e.g., face models and skin color) for face detection and tracking [89]. A method of combining a Hidden Markov Model (HMM) and motion for tracking was presented in [86]. A combination of motion and color filters, and a neural network model was proposed in [19].

(a) (b) (c) (d)

Figure 2.1. Outputs of several face detection algorithms; (a), (b) Féraud et al. [19]; (c) Maio et al. [20]; (d) Garcia et al. [21]; (e), (f) Schneiderman et al. [22]; (g) Rowley et al. [23]; (h), (i) Rowley et al. [24]; (j) Sung et al. [25]; (k) Yow et al. [26]; (l) Lew et al. [27].



(e) (f) (g) (h)



(i) (j) (k) (l)

Figure 2.1. (Cont'd).

Categorizing face detection methods based on their representations of faces reveals that detection algorithms using holistic representations have the advantage of finding small faces or faces in poor-quality images, while those using geometrical facial features provide a good solution for detecting faces in different poses. A combination of holistic and feature-based methods [59], [60] is a promising approach to face detection as well as face recognition. Motion [86], [19] and skin-tone color [19], [84], [90], [85], [21] are useful cues for face detection. However, the color-based approaches face difficulties in robustly detecting skin colors in the presence of complex background and variations in lighting conditions. Two color spaces (YC_bC_r and HSV) have been proposed for detecting the skin color patches to compensate for lighting variations [21]. We propose a face detection algorithm that is able to handle a wide range of color variations in static images, based on a lighting compensation technique in the RGB color space and a nonlinear color transformation in the YC_bC_r color space. Our approach models skin color using a parametric ellipse in a two-dimensional transformed color space and extracts facial features by constructing feature maps for the eyes, mouth and face boundary from color components in the YC_bC_r space.

2.2 Face Recognition

The human face has been considered as the most informative organ for communication in our social lives [49]. Automatically recognizing faces by machines can facilitate a wide variety of forensic and security applications. The representation of human faces for recognition can vary from a 2D image to a 3D surface. Different representations

result in different recognition approaches. Extensive reviews of approaches to face recognition were published in 1995 [37], 1999 [31], and in 2000 [38]. A workshop on face processing in 1985 [91] presented studies of face recognition mainly from the viewpoint of cognitive psychology. Studies of feature-based face recognition, computer caricatures, and the use of face surfaces in simulation and animation were summarized in 1992 [49]. In 1997, Uwechue et al. [92] gave details of face recognition based on high-order neural networks using 2D face patterns. In 1998, lectures on face recognition using 2D face patterns were presented from theory to applications [36]. In 1999, Hallinan et al. [93] described face recognition using both the statistical models for 2D face patterns and the 3D face surfaces. In 2000, Gong et al. [94] emphasized the statistical learning methods in holistic recognition approaches and discussed face recognition from the viewpoint of dynamic vision.

The above studies show that the face recognition techniques, especially holistic methods based on the statistical pattern theory, have greatly advanced over the past ten years. Face recognition systems (e.g., FaceIt [1] and FaceSnap [2]) are being used in video surveillance and security monitoring applications. However, more reliable and robust techniques for face recognition as well as detection are required for several applications. Except for the recognition applications based on static frontal images that are taken under well-controlled environments (e.g., indexing and searching large image database of drivers for issuing driving licenses), the main challenge in face recognition is to be able to deal with the high degree of variability in human face images. The sources of variations include inter-subject variations (distinctiveness of individual appearance) and intra-subject variations (in 3D pose, facial expression,

facial hair, lighting, and aging). Some variations are not removable, while others can be compensated for recognition. Persons who have similar face appearances, e.g. twins, and an individual who could have different appearances due to cosmetics, or other changes in facial hair and glasses are very difficult to recognize. Variations due to different poses, illuminations, and facial expressions are *relatively easy* to handle. Currently available algorithms for face recognition concentrate on recognizing faces under those variations which can somehow be compensated for. Because facial variations due to pose cause a large amount of appearance change, more and more systems are taking advantage of 3D face geometry for recognition.

The performance of a recognition algorithm depends on the face databases it is evaluated on. Several face databases, such as MIT [95], Yale [96], Purdue [97], and Olivetti [98] databases are publically available for researchers. Figure 2.2 shows some examples of face images from the FERET [28], MIT [29], and XM2VTS [30] databases. According to Phillips [68], [28], the FERET evaluation of face recognition algorithms identifies three state-of-the-art techniques: (i) the principal component analysis (PCA)-based approach [99], [100], [29]; (ii) the elastic bunch graphic matching (EBGM)-based paradigm [32]; and (iii) the local feature analysis (LFA)-based approach [34], [101]. The internal representations of PCA-based, EBGM-based, and LFA-based recognition approaches are shown in Figs. 2.3, 2.4, and 2.5, respectively. To represent and match faces, the PCA-based approach makes use of a set of orthonormal basis images; the EBGM-based approach constructs a face bunch graph, whose nodes are associated with a set of wavelet coefficients (called jets); the LFA-based approach uses localized kernels, which are constructed from PCA-based eigenvectors,

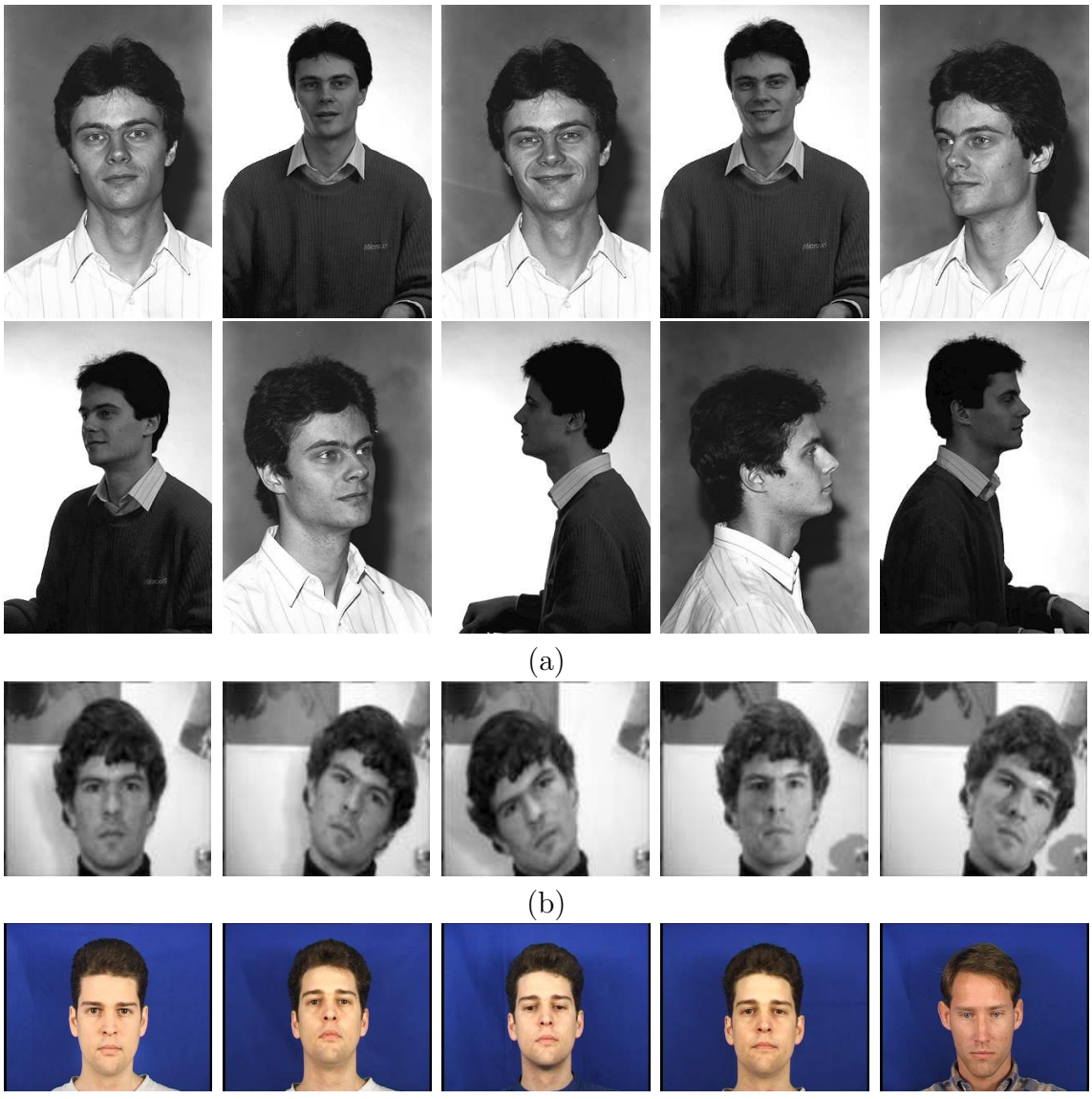


Figure 2.2. Examples of face images are selected from (a) the FERET database [28]; (b) the MIT database [29]; (c) the XM2VTS database [30].

for topographic facial features (e.g., eyebrows, cheek, mouth, etc.)

Mean MEF1 MEF2 MEF3 MEF4 MEF5 MEF6 MEF7 MEF8

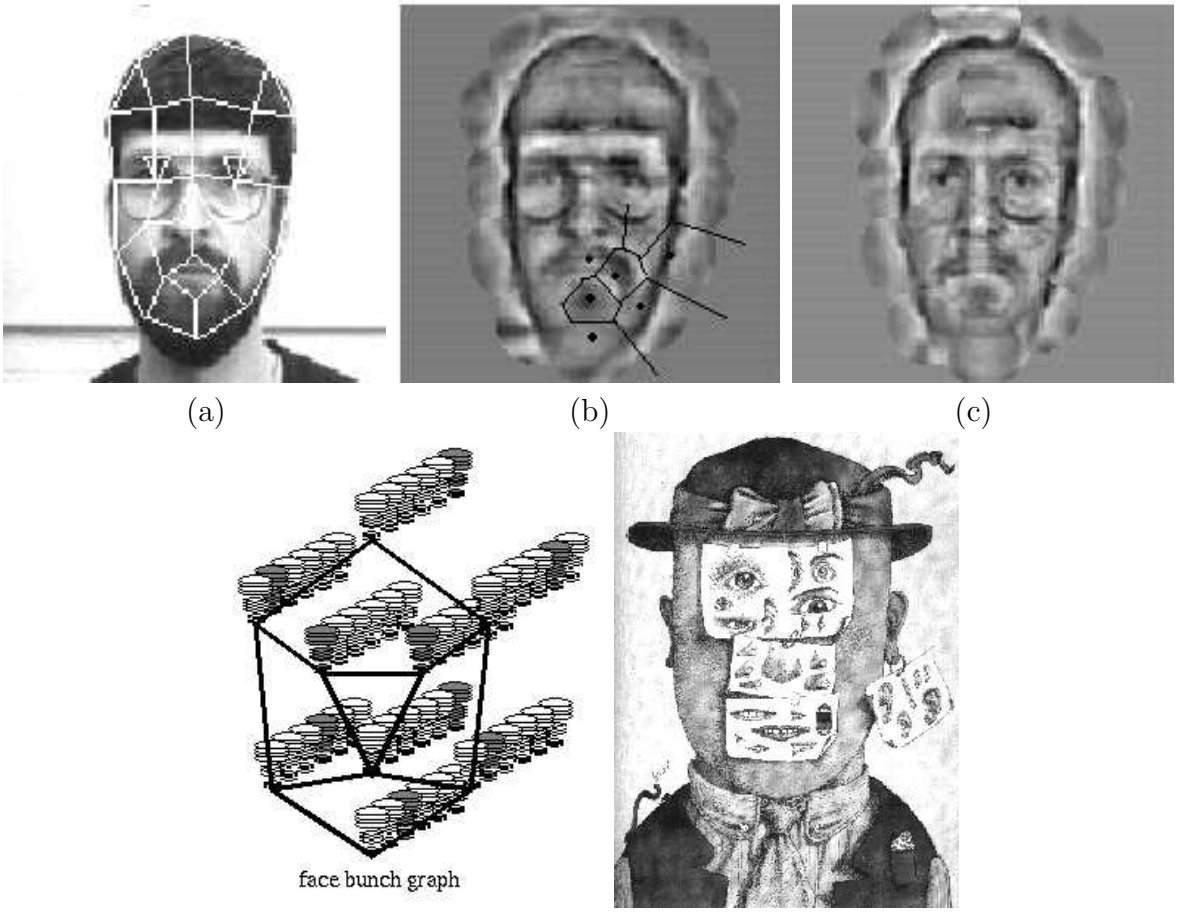
(a)

Mean MDF1 MDF2 MDF3 MDF4 MDF5 MDF6 MDF7 MDF8

(b)

Figure 2.3. Internal representations of the PCA-based approach and the LDA-based approach (from Weng and Swets [31]). The average (mean) images are shown in the first column. Most Expressive Features (MEF) and Most Discriminating Features (MDF) are shown in (a) and (b), respectively.

The PCA-based algorithm provides a compact but non-local representation of face images. Based on the appearance of an image at a specific view, the PCA algorithm works at the pixel level. Hence, the algorithm can be regarded as "picture" recognition, in other words, it is not explicitly using any facial features. The EBGM-based algorithm constructs local features (extracted using Gabor wavelets) and global face shape (represented as a graph), and so this approach is much closer to "face" recognition. However, the EBGM algorithm is pose-dependent, and it requires initial graphs for different poses during its training stage. The LFA-based algorithm is derived from the PCA-based method; it is also called a kernel PCA method. In this approach, however, the choice of kernel functions for local facial features (e.g., eyes, mouth, and nose) and the selection of locations of these features still remains an open



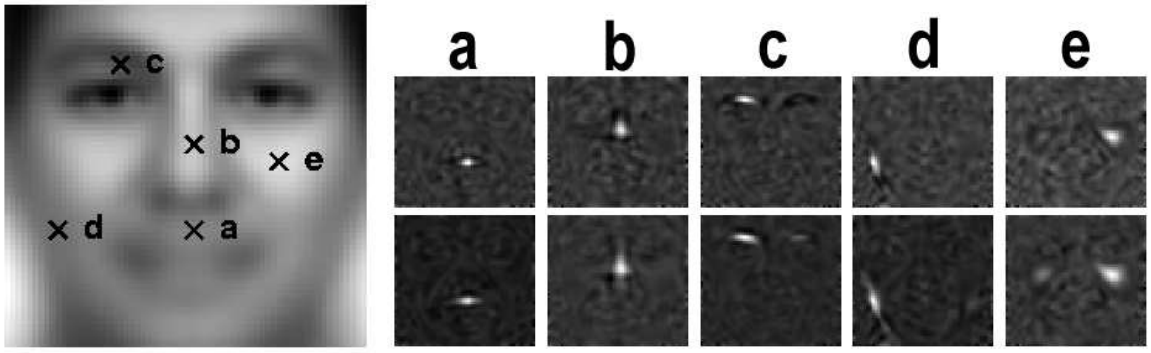
(a) (b) (c)

(d) (e)

Figure 2.4. Internal representations of the EBGM-based approach (from Wiskott et al. [32]): (a) a graph is overlaid on a face image; (b) a reconstruction of the image from the graph; (c) a reconstruction of the image from a face bunch graph using the best fitting jet at each node. Images are downloaded from [33]; (d) a bunch graph whose nodes are associated with a bunch of jets [33]; (e) an alternative interpretation of the concept of a bunch graph [33].

question.

In addition to these three approaches, we categorize face recognition algorithms on the basis of pose-dependency and matching features (see Fig. 2.6). In pose-dependent algorithms, a face is represented by a set of viewer-centered images. A small number of 2D images (appearances) of a human face at different poses are stored as a representative set of the face, while the 3D face shape is implicitly represented in the set.



(a) (b)

Figure 2.5. Internal representations of the LFA-based approach (from Penev and Atick [34]). (a) An average face image is marked with five localized features; (b) five topographic kernels associated with the five localized features are shown in the top row, and the corresponding residual correlations are shown in the bottom row.

The representative set can be obtained from either digital cameras or extracted from videos. On the other hand, in pose-invariant approaches, a face is represented by a 3D face model. The 3D face shape of an individual is explicitly represented, while the 2D images are implicitly encoded in this face model. The 3D face models can be constructed by using either 3D digitizers or range sensors, or by modifying a generic face model using a video sequence or still face images of frontal and profile views.

The pose-dependent algorithms can be further divided into three classes: appearance-based (holistic) [29], [78] feature-based (analytic) [102], [103] and hybrid (which combines holistic and analytic methods) [60], [99], [32], [34] approaches. The appearance-based methods are sensitive to intra-subject variations, especially to changes in hairstyle, because they are based on global information in an image. However, the feature-based methods suffer from the difficulty of detecting local fiducial "points". The hybrid approaches were proposed to accommodate both global and local face shape information. For example, LFA-based methods, eigen-template meth-

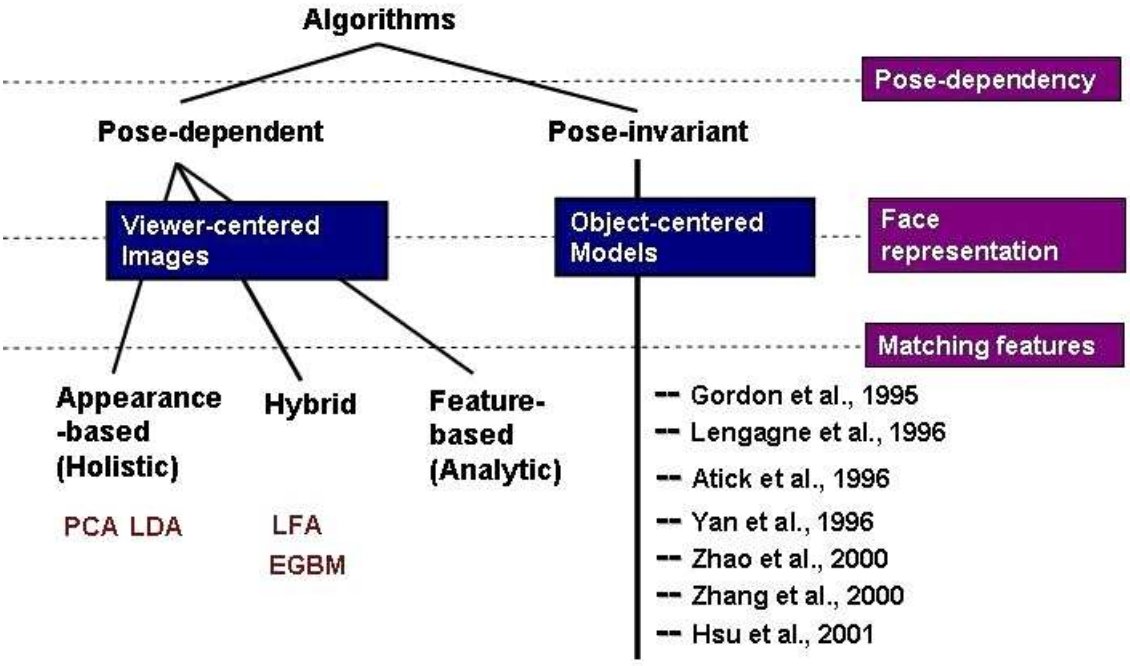


Figure 2.6. A breakdown of face recognition algorithms based on the pose-dependency, face representation, and features used in matching.

ods, and shape-and-shape-free [104] methods belong to the hybrid approach which is derived from the PCA methodology. The EBGM-based methods belong to the hybrid approach that is based on 2D face graphs and wavelet transforms at each feature node of the graphs. Although they are in the hybrid approach category, the eigen-template matching and EBGM-based methods are much closer to feature-based approaches.

In the pose-invariant algorithms, 3D face models are utilized to reduce the variations in pose and illumination. Gordon et al. [105] proposed an identification system based on 3D face recognition. The 3D model used by Gordon et al. is represented by a number of 3D points associated with their corresponding texture features. This method requires an accurate estimate of the face pose. Lengagne et al. [106] proposed a 3D face reconstruction scheme using a pair of stereo images for recognition and mod-

eling. However, they did not implemented the recognition module. Atick et al. [107] proposed a reconstruction method of 3D face surfaces based on the Karhonen-Loeve (KL) transform and the shape-from-shading approach. They discussed the possibility of using *eigenhead surfaces* in face recognition applications. Yan et al. [108] proposed a 3D reconstruction method to improve the performance of face recognition by making Atick et al.'s reconstruction method rotation-invariant. Zhao et al. [109] proposed a method to adapt a 3D model from a generic range map to the shape obtained from shading for enhancing face recognition performance in different lighting and viewing conditions.

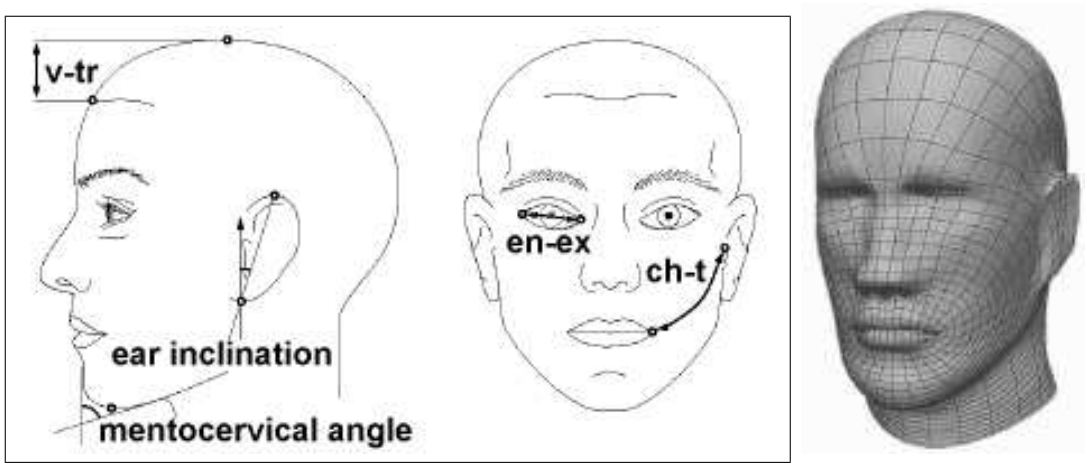
Based on our brief review, we believe that the current trend is to use 3D face shape explicitly for recognition. In order to efficiently store an individual's face, one approach is to adapt a 3D face model [72] to the individual. There is still a considerable debate on whether the internal recognition mechanism of a human brain involves explicit 3D models or not [49], [110]. However, there is enough evidence to support the fact that humans use information about 3D structure of objects (e.g., 3D geometry of a face) for recognition. Closing our eyes and imagining a face (or a chair) can easily verify this hypothesis, since the structure of a face (or a chair) can appear in our mind without the use of eyes. Moreover, the use of a 3D face model can separate both geometrical and texture features for facial analysis, and can also blend both of them for recognition as well as visualization [67]. Our proposed systems belong to this emerging trend.

2.3 Face Modeling

Face modeling plays a crucial role in applications such as human head tracking, facial animation, video compression/coding, facial expression recognition, and face recognition. Researchers in computer graphics have been interested in modeling human faces for facial animation. Applications such as virtual reality and augmented reality [74] require modeling faces for human simulation and communication. In applications based on face recognition, modeling human faces can provide an explicit representation of a face that aligns facial shape and texture features together for face matching at different poses and in different illumination conditions.

2.3.1 Generic Face Models

We first review three major approaches to modeling human faces and then point out an advanced modeling approach that makes use of the *a priori* knowledge of facial geometry. DeCarlo et al. [111] use the anthropometric measurements to generate a general face model (see Fig. 2.7). This approach starts with manually-constructed B-spline surfaces and then applies surface fitting and constraint optimization to these surfaces. It is computationally intensive due to its optimization mechanism. In the second approach, facial measurements are directly acquired from 3D digitizers or structured light range sensors. 3D models are obtained after a postprocessing, triangularization, on these shape measurements. The third approach, in which models are reconstructed from photographs, only requires low-cost and passive input devices (video cameras). Some computer vision techniques for reconstructing 3D data can

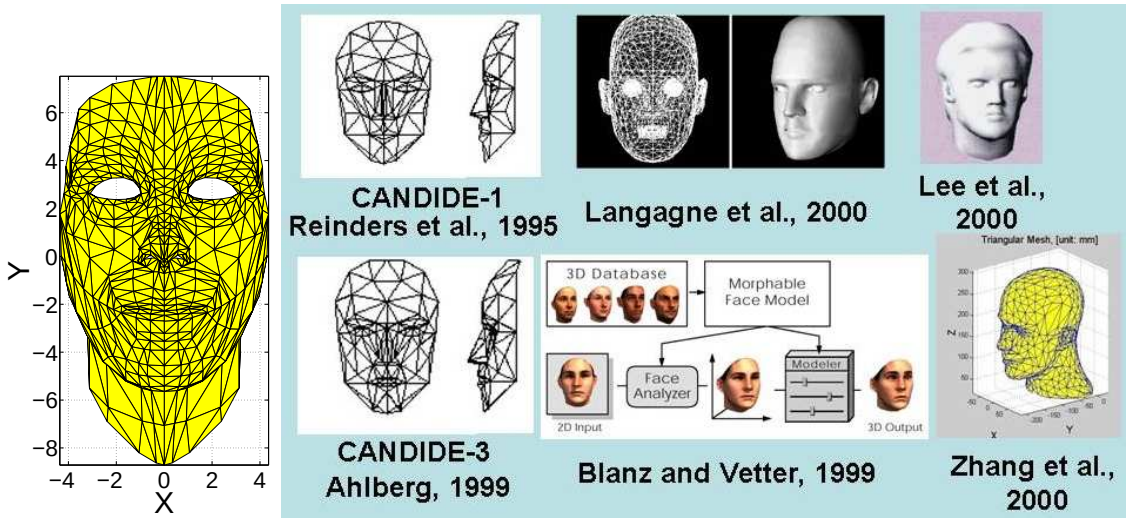


(a) (b)

Figure 2.7. Face modeling using anthropometric measurements (downloaded from [35]): (a) anthropometric measurements; (b) a B-spline face model.

be used for face modeling. For instance, Lengagne et al. [106] and Chen et al. [112] built face models from a pair of stereo images. Atick et al. [107] and Yan et al. [108] reconstructed 3D face surfaces based on the Karhonen-Loeve (KL) transform and the shape-from-shading technique. Zhao et al. [109] made use of a symmetric shape-from-shading technique to build a 3D face model for recognition. There are other methods which combine both shape-from-stereo (which extracts low-spatial frequency components of 3D shape) and shape-from-shading (extracting high-spatial frequency components) to reconstruct 3D faces [113], [114], [115]. See [116] for additional methods to obtain facial surface data. However, currently it is still difficult to extract sufficient information about the facial geometry only from 2D images. This difficulty is the reason why Guenter et al. [117] utilize a large number of fiducial points to capture 3D face geometry for photorealistic animation. Even though we can obtain dense 3D facial measurements from high-cost 3D digitizers, it takes too much time and it is expensive to scan a large number of human subjects.

An advanced modeling approach which incorporates *a priori* knowledge of facial geometry has been proposed for efficiently building face models. We call the model representing the general facial geometry as a generic face model. Waters' face model [69], shown in Fig. 2.8(a), is a well-known instance of polygonal facial surfaces. Figure 2.8(b) shows some other generic face models. The one used by Blanz and Vet-



(a) (b)

Figure 2.8. Generic face models: (a) Water's animation model; (b) anthropometric measurements; (b) six kinds of face models for representing general facial geometry.

ter is a statistics-based face model which is represented by the principal components of shape and texture data. Reinders et al. [72] used a fairly *coarse* wire-frame model, compared to Waters' model, to do model adaptation for image coding. Yin et al. [118] proposed a MPEG4 face modeling method that uses fiducial points extracted from two face images at frontal and profile views. Their feature extraction is simply based on the results of intensity thresholding and edge detection. Similarly, Lee et al. [119] have proposed a method that modifies a generic model using either two orthogonal pictures (frontal and profile views) or range data, for animation. Similarly,

for facial animation, Lengagne et al. [120] and Fua [121] use bundle-adjustment and least-squares fitting to fit a complex animation model to uncalibrated videos. This algorithm makes use of stereo data, silhouette edges, and 2D feature points. Five manually-selected features points and initial values of camera positions are essential for the convergence of this method. Ahlberg [122] adapts a 3D wireframe model (CANDIDE-3 [123]) to a 2D video image. The two modeling methods proposed in this thesis follow the modeling approach using a generic face model; both of our methods make use of a generic face model (Waters' face model) as *a priori* knowledge of facial geometry and employ (i) displacement propagation and 2.5D snakes in the first method and (ii) interacting snakes and semantic face graphs in the second method for adapting recognition-orientated features to an individual's geometry.

2.3.2 Snakes for Face Alignment

As a computational bridge between the high-level a priori knowledge of object shape and the low-level image data, snakes (or active contours) are useful models for extracting the shape of deformable objects. Similar to other template-based approaches such as Hough transform and active shape models, active contours have been employed to detect object boundary, track objects, reconstruct 3D objects (stereo snakes and inter-frame snakes), and match/identify shape. Snakes self converge in an iterative way, and deform either with or without topological constraints.

Research on active contours focuses on issues related to representation (e.g., parametric curves, splines, Fourier series, and implicit level-set functions), energy func-

tionals to minimize, implementation methods (e.g., classical finite difference models, dynamic programming [124], and Fourier spectral methods), convergence rates and conditions, and their relationship to statistical theory [125] (e.g., the Bayesian estimation). Classical snakes [77], [126] are represented by parametric curves and are deformed by finite difference methods based on edge energies. In applications, different types of edge energies including image gradients, gradient vector flows [127], distance maps, and balloon force have been proposed. On the other hand, combined with level-set methods and the curve evolution theory, active contours have emerged as a powerful tool, called geodesic active contours (GAC) [128], to extract deformable objects with unknown geometric topology. However, in the GAC approach, the contours are implicitly represented as level-set functions and are closed curves. In addition to the edge energy, region energy has been introduced to improve the segmentation results for homogeneous objects in both the parametric and the GAC approaches (e.g., region and edge [129], GAC without edge [130], statistical region snake [131], region competition [132], and active region model [133]). Recently, multiple active contours [134], [135] were proposed to extract/partition multiple homogeneous regions that do not overlap with each other in an image.

In our first alignment method, we have reformulated 2D active contours (a dynamic programming approach) in 3D coordinates for energies derived from 2.5D range and 2D color data. In our second alignment method, we make use of multiple 2D snakes (a finite difference approach) that interact with each other in order to adapt facial components.

2.3.3 3D Model Compression

Among various representations of 3-D objects, surface models can explicitly represent shape information and can effectively provide a visualization of these objects. The polygonal model using triangular meshes is the most prevalent type of surface representations for free-form objects such as human faces. The reason is that the mesh model explicitly describes the connectivity of surfaces, enables mesh simplification, and is suitable for free-form objects [136]. The polygonization of an object surface approximates the surface by a large number of triangles (facets), each of which contains primary information about vertex positions as well as vertex associations (indices), and auxiliary information regarding facet properties such as color, texture, specularity, reflectivity, orientation, and transparency. Since we use a triangular mesh to represent a generic face model and an adapted model, model compression is preferred when efficient transmission, visualization, and storage is required.

In 1995, the concept of geometric compression was first introduced by Deering [137], who proposed a technique for lossy compression of 3-D geometric data. Deering's technique focuses mainly on the compression of vertex positions and facet properties of 3-D triangle data. Taubin [75] proposed *topological surgery* which further contributed connectivity encoding (compression of association information) to geometric compression. Lounsbery et al. [76] performed geometric compression through multiresolution analysis for particular meshes with subdivision connectivity. Applying remeshing algorithms to arbitrary meshes, Eck et al. [138] extended Lounsbery's work on mesh simplification. Typical compression ratios in this line of development

are listed in Table 2.2. All of these compression methods focus on model represen-

Table 2.2
Geometric compression efficiency.

Method	Geometric Compression Ratio (GCR)	Loss Measure	Compressed feature
Geometric Compression [137]	6–10	slight losses	Positions, normals, colors
Topological Surgery [75]	20–100 12–30 20–100	no loss N/A N/A	Connectivity; Positions, facet properties; ASCII-file sizes
Remeshing [138]	54–1.2	Remeshing & compression tolerances	Level of detail (facets)

tation using triangular meshes. However, for more complex 3D shapes, the surface representation using triangular meshes usually results in a large number of triangular facets, because each triangular facet is explicitly described. We have developed a novel compression approach for free-form surfaces using 3D wavelets and lattice vector quantization [139]. In our approach, surfaces are implicitly represented inside a volume in the same way as edges in a 2D image. A further improvement in our approach can be achieved by making use of integer wavelet transformation [140], [141].

2.4 Face Retrieval

Face recognition technology provides a useful tool for content-based image and video retrieval using the concept of human faces. Based on face detection and identification technology, we can design a system for consumer photo management (or for web graphic search) that uses human faces for indexing and retrieving image content and generates annotation (textual descriptions) for the image content automatically.

Traditional text-based retrieval systems for digital libraries can not fulfill a retrieval of visual content such as human faces, eye shape, and cars in image or video databases. Hence, many researchers have been developing multimedia retrieval techniques based on automatically extracting salient features from the visual content (see [40] for an extensive review). Well known systems for content-based image and video retrieval are QBIC [142], Photobook [143], CONIVAS [144], FourEyes [145], Virage [146], ViBE [147], VideoQ [148], Visualseek [149], Netra [150], MARS [151], PicSOM [152], ImageScape [153], etc. In these systems, retrieval is performed by comparing a set of low-level features of a query image or video clip with features stored in the database and then by presenting the user with the content that has the most similar features. However, users normally query an image or video database based on semantics rather than low-level features. For example, a typical query might be specified as "retrieve images of fireworks" rather than "retrieve images that have large dark regions and colorful curves over the dark regions".

Since the commonly used features are usually a set of unorganized low-level attributes (such as color, texture, geometrical shape, layout, and motion), grouping

low-level features can provide meaningful high-level semantics for human consumers. There has been some work done on automatically classifying images into semantic categories [154], such as indoors/outdoors and city/landscape images. As for the semantic concept of faces, the generic facial topology (e.g., our proposed generic semantic face graph) is a useful structure for representing the **face** in a search engine. We have designed a graphical user interface for face editing using our face detection algorithm. Combined with our semantic face matching algorithm, we can build a face retrieval system.

2.5 Summary

We have briefly described the development of face detection, face recognition, face modeling and model compression in this chapter. We have summarized the performance of currently available face detection systems in Table 2.3. Note that the performance of a detection system depends on several factors such as face databases on which the system is evaluated, system architecture, distance metric, and algorithmic parameters. The performance is evaluated based on the detection rate, the false positive rate (false acceptance rate), and databases. In Table 2.3, we do not include the false acceptance rate because the false positive rate has not been completely reported in literature. We refer the reader to the FERET evaluation [68], [28] for the performance of various face recognition systems.

Face detection and face recognition are closely related to each other in the sense of categorizing faces. Over the past ten years, based on the statistical pattern theory,

Table 2.3
Summary of performance of various face detection approaches.

Authors	Year	Head Pose	Test Databases	Detection Rate
Féraud et al. [19]	2001	Frontal to profile	Sussex; CMU test1; Web images	100% for Sussex; 81% $\sim$ 86% for CMU test1; 74.7% $\sim$ 80.1% for Web images.
Maio et al. [20]	2000	Frontal	Static images	89.53% $\sim$ 91.34%
Schneiderman et al. [22]	2000	Frontal to profile	CMU; Web images	75.24% $\sim$ 92.7%
Garcia et al. [21]	1999	Frontal to near frontal	MPEG videos	93.27%
Rowley et al. [24], [23]	1998	(Upright) frontal	CMU; FERET; Web images	86%[24]; 79.6%[23] for rotated faces
Yow et al. [26]	1997	Frontal to profile	CMU	84% $\sim$ 92%
Lew et al. [27]	1996	Frontal	MIT; CMU; Leiden	87% $\sim$ 95%

the appearance-based (holistic) approach has greatly advanced the field of face recognition. By categorizing face detection methods based on their representations of the face, we observe that detection/recognition algorithms using holistic representations have the advantage of finding/identifying small faces or faces in poor-quality images (i.e. detection/recognition under uncertainty), while those using geometrical facial features provide a good solution for detecting/recognizing faces in different poses and expressions. The internal representation of a human face substantially affects the performance and design of a detection or recognition system. A seamless combination of

holistic 2D and geometrical 3D features provides a promising approach to represent faces for face detection as well as face recognition. Modeling human face in 3D space has been shown to be useful for face recognition. However, the important aspect of face modeling is how to *efficiently encode* the 3D facial geometry and texture as compact features for face recognition.

Chapter 3

Face Detection

We will first describe an overview of our proposed face detection algorithm and then give details of the algorithm. We will demonstrate the performance and experimental results on several image databases.

3.1 Face Detection Algorithm

The use of color information can simplify the task of face localization in complex environments [19], [84], [90], [85]. Therefore, we use skin color detection as the first step in detecting faces. An overview of our face detection algorithm is depicted in Fig. 3.1, which contains two major modules: (i) face localization for finding face candidates; and (ii) facial feature detection for verifying detected face candidates. The face localization module combines the information extracted from the luminance and the chrominance components of color images and some heuristics about face shape (e.g., face sizes ranging from 13×13 pixels to about three fourths of the image

size) to generate potential *face candidates*, within the entire image. The algorithm first estimates and corrects the color bias based on a novel lighting compensation technique. The corrected red, green, and blue color components are first converted to the YC_bC_r color space and then nonlinearly transformed in this color space (see formulae in Appendix A). The skin-tone pixels are detected using an elliptical skin model in the transformed space. The parametric ellipse corresponds to contours of constant Mahalanobis distance under the assumption of the Gaussian distribution of skin tone color. The detected skin-tone pixels are iteratively segmented using local color variance into connected components which are then grouped into face candidates based on both the spatial arrangement of these components (described in Appendix B) and the similarity of their color [84]. Figure 3.1 shows the input color image, color compensated image, skin regions, grouped skin regions, and face candidates obtained from the face localization module. Each grouped skin region is assigned a pseudo color and each face candidate is represented by a rectangle. Because multiple face candidates (bounding rectangles) usually overlap, they can be fused based on the percentage of overlapping areas. However, in spite of this postprocessing there are still some false positives among face candidates.

It is inevitable that detected skin-tone regions will include some non-face regions whose color is similar to the skin-tone. The facial feature detection module rejects face candidate regions that do not contain any facial features such as eyes, mouth, and face boundary. This module can detect multiple eye and mouth candidates. A triangle is constructed from two eye candidates and one mouth candidate, and the best-fitting enclosing ellipse of the triangle is constructed to approximate the face

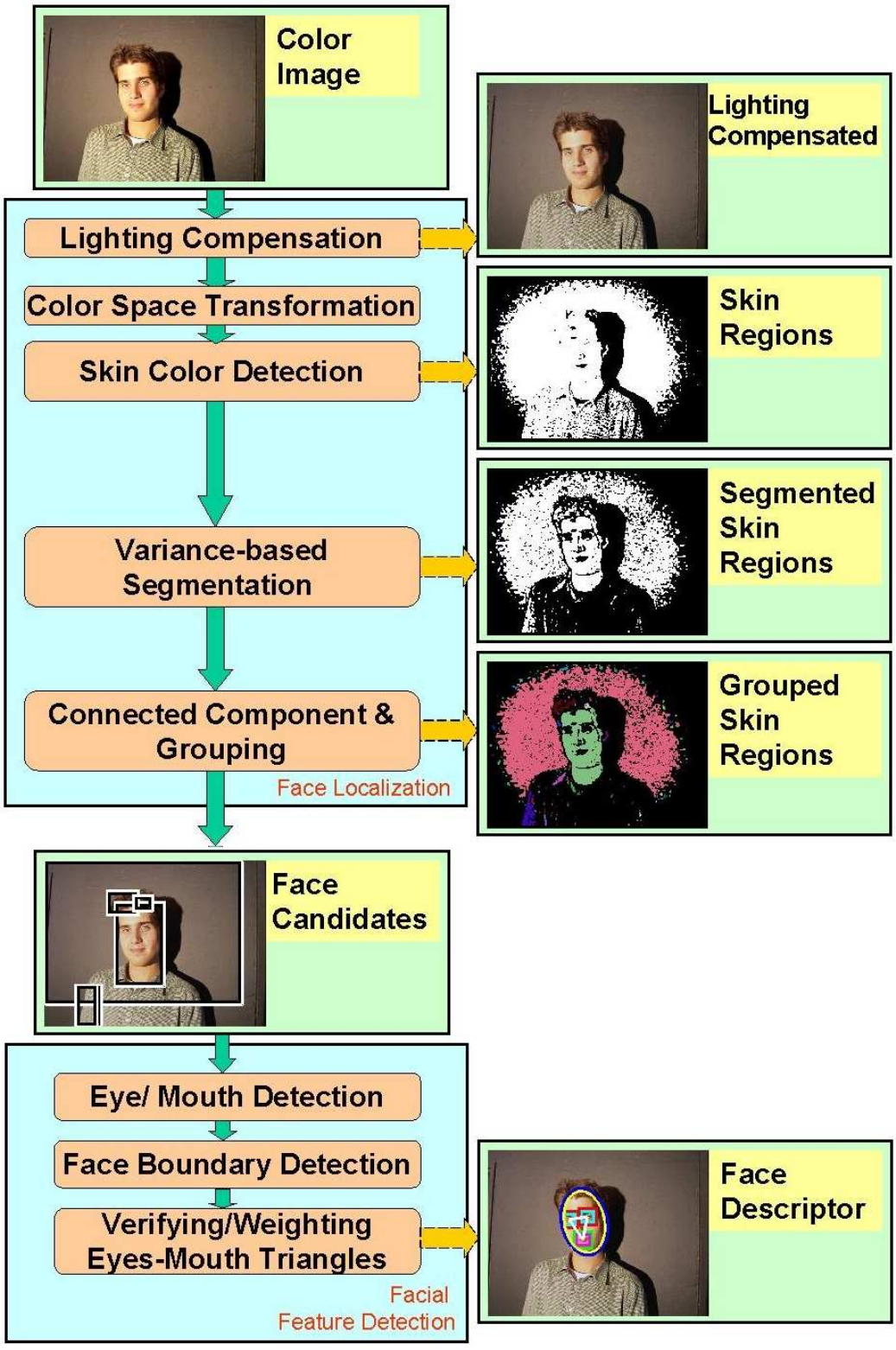


Figure 3.1. Face detection algorithm. The face localization module finds face candidates, which are verified by the detection module based on facial features.

boundary. A face score is computed for each set of eyes, mouth and the ellipse. Figure 3.1 shows a detected face and the enclosing ellipse with its associated eye-mouth triangle which has the highest score that exceeds a threshold. These detected facial features are grouped into a structured facial descriptor in the form of a 2D graph for face description. These descriptors can be the input to subsequent modules such as face modeling and recognition. We now describe in detail the individual components of the face detection algorithm.

3.2 Lighting Compensation and Skin Tone Detection

The appearance of the skin-tone color can change due to different lighting conditions. We introduce a lighting compensation technique that uses "reference white" to normalize the color appearance. We regard pixels with the top 5% of the luma (nonlinear gamma-corrected luminance) values as the reference white if the number of these pixels is sufficiently large (> 100). The red, green, and blue components of a color image are adjusted so that these reference-white pixels are scaled to the gray level of 255. The color components are unaltered if a sufficient number of reference-white pixels is not detected. This assumption is reasonable not only because an image usually contains "real white" (i.e., white reference in [155]) pixels in some regions of interest (such as eye regions), but also because the dominant bias color always appears in the "real white". Figure 3.2 demonstrates an example of our lighting compensation

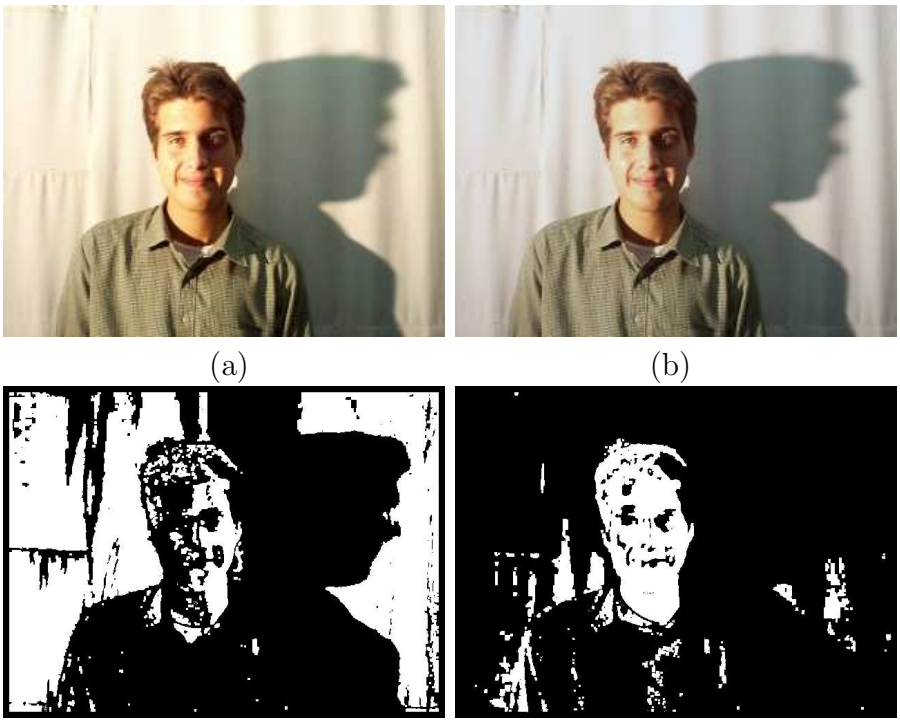


Figure 3.2. Skin detection: (a) a yellow-biased face image; (b) a lighting compensated image; (c) skin regions of (a) shown in white; (d) skin regions of (b).

method. Note that the yellow bias color in Fig. 3.2(a) has been removed, as shown in Fig. 3.2(b). The effect of lighting compensation on detected skin regions can be seen by comparing Figs. 3.2(c) and 3.2(d). With lighting compensation, our algorithm detects fewer non-face areas and more skin-tone facial areas. Note that the variations in skin color among different racial groups, reflection characteristics of human skin and its surrounding objects (including clothing), and camera characteristics will all affect the appearance of skin color and hence the performance of an automatic face detection algorithm. Therefore, if models of the lighting source and cameras are available, additional lighting correction should be made to remove color bias.

Modeling skin color requires choosing an appropriate color space and identifying a cluster associated with skin color in this space. It has been observed that the

normalized red-green (r-g) space [156] is not the best choice for face detection [157], [158]. Based on Terrillon et al.'s [157] comparison of nine different color spaces for face detection, the tint-saturation-luma (TSL) space provides the best results for two kinds of Gaussian density models (unimodal and mixture of Gaussian densities). We adopt the YC_bC_r space since it is perceptually uniform [155], is widely used in video compression standards (e.g., MPEG and JPEG) [21], and it is similar to the TSL space in terms of the separation of luminance and chrominance as well as the compactness of the skin cluster. Many research studies assume that the chrominance components of the skin-tone color are independent of the luminance component [159], [160], [158], [90]. However, in practice, the skin-tone color is nonlinearly dependent on luminance. In order to demonstrate the luma dependency of skin-tone color, we manually collected training samples of skin patches ($853,571$ pixels) from 9 subjects (137 images) in the Heinrich-Hertz-Institute (HHI) image database [15]. These pixels form an elongated cluster that shrinks at high and low luma in the YC_bC_r space, shown in Fig. 3.3(a). Detecting skin tone based on the cluster of training samples in the C_b-C_r subspace, shown in Fig. 3.3(b), results in many false positives. If we base the detection on the cluster in the (C_b/Y)-(C_r/Y) subspace, shown in Fig. 3.3(c), then many false negatives result. The dependency of skin tone color on luma is also present in the normalized rgY space in Fig. 3.4(a), the perceptually uniform $CIE\ xyY$ space in Fig. 3.4(c), and the HSV spaces in Fig. 3.4(e). The 3D cluster shape changes at different luma values, although it looks compact in the 2D projection subspaces, in Figs. 3.4(b), 3.4(d) and 3.4(f).

To deal with the skin-tone color dependence on luminance, we nonlinearly trans-

form the YC_bC_r color space to make the skin cluster luma-independent. This is done by fitting a piecewise linear boundary to the skin cluster (see Fig. 3.5). The details of the model and the transformation are described in Appendix A. The transformed space, shown in Fig. 3.6(a), enables a robust detection of dark and light skin tone colors. Figure 3.6(b) shows the projection of the 3D skin cluster in the transformed C_b-C_r color subspace, on which the elliptical model of skin color is overlaid. Figure 3.7 shows examples of detection using the nonlinear transformation. More skin-tone pixels with low and high luma are detected in this transformed subspace than in the C_bC_r subspace.

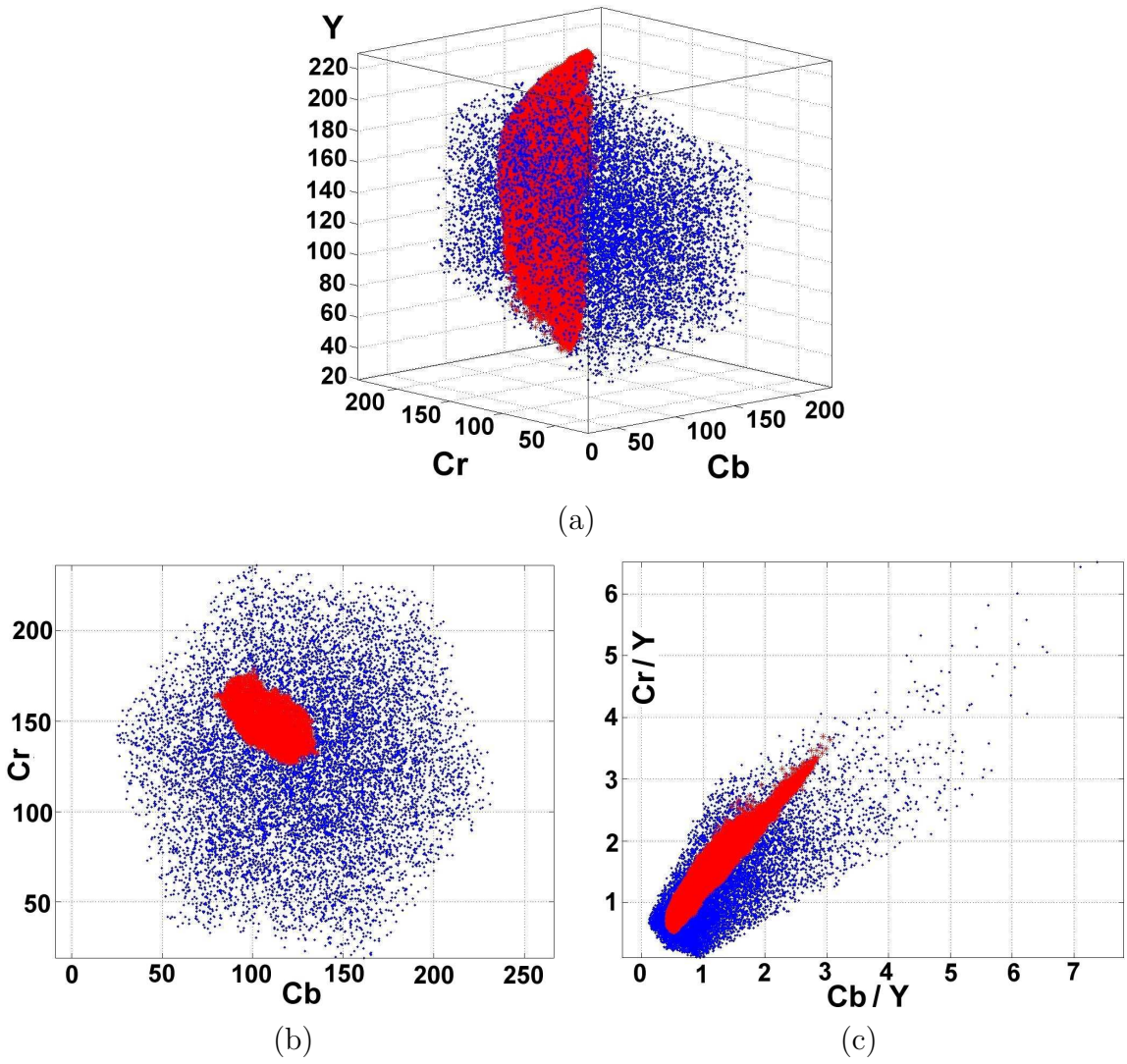


Figure 3.3. The YC_bC_r color space (blue dots represent the reproducible color on a monitor) and the skin tone model (red dots represent skin color samples). (a) The YC_bC_r space; (b) a 2D projection in the C_b-C_r subspace; (c) a 2D projection in the (C_b/Y)-(C_r/Y) subspace.

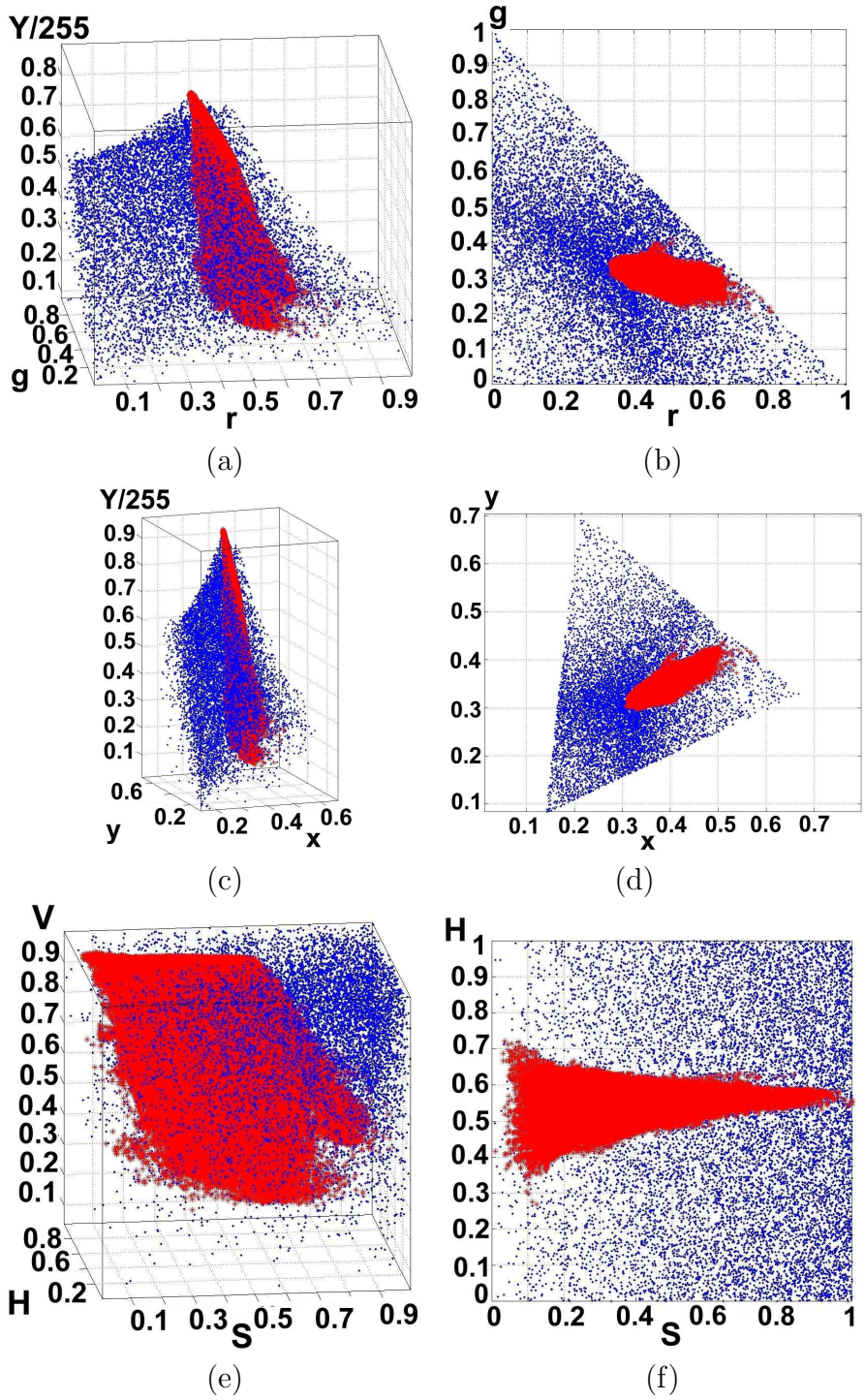
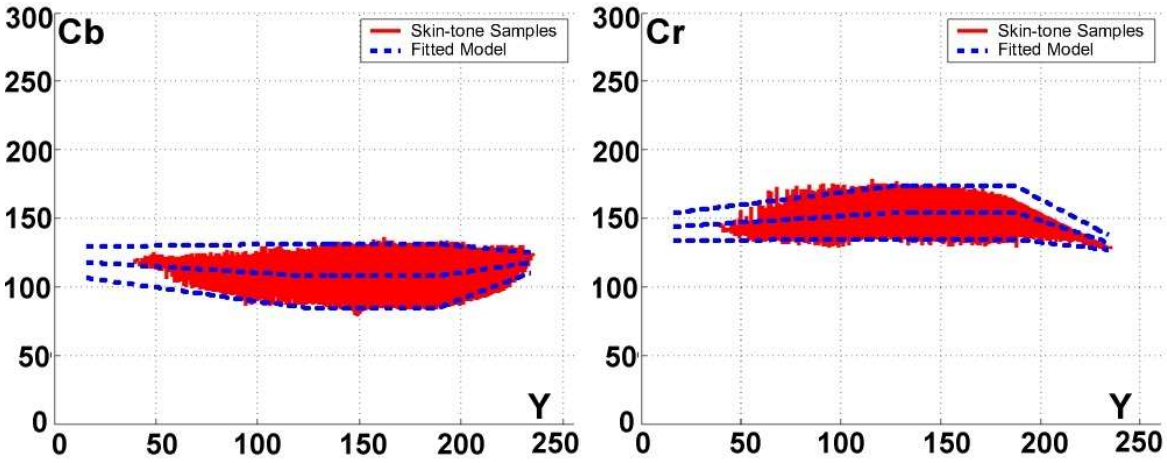
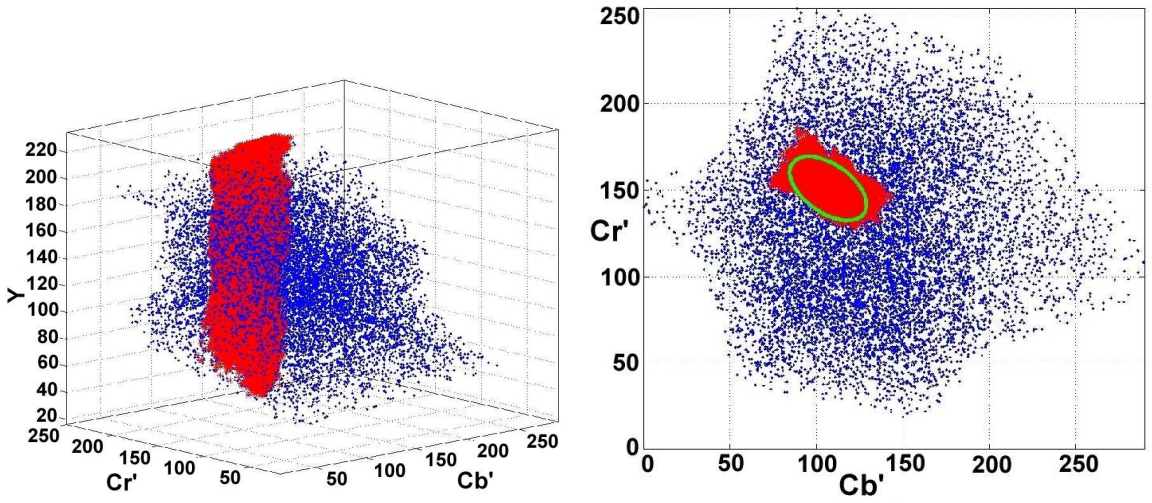


Figure 3.4. The dependency of skin tone color on luma. The skin tone cluster (red dots) is shown in (a) the rgY, (c) the $CIE\ xyY$, and (e) the HSV color spaces; the 2D projection of the cluster is shown in (b) the $r-g$, (d) the $x-y$, and (f) $S-H$ color subspaces, where blue dots represent the reproducible color on a monitor. For a better presentation of cluster shape, we normalize the luma Y in the rgY and the $CIE\ xyY$ by 255, and swap the hue and saturation coordinates in the HSV space. The skin tone cluster is less compact at low saturation values in (e) and (f).



(a) (b)

Figure 3.5. 2D projections of the 3D skin tone cluster in (a) the Y-C_b subspace; (b) the Y-C_r subspace. Red dots indicate the skin cluster. Three blue dashed curves, one for cluster center and two for boundaries, indicate the fitted models.



(a) (b)

Figure 3.6. The nonlinear transformation of the YC_bC_r color space. (a) The transformed YC_bC_r color space; (b) a 2D projection of (a) in the C_b-C_r subspace, in which the elliptical skin model is overlaid on the skin cluster.

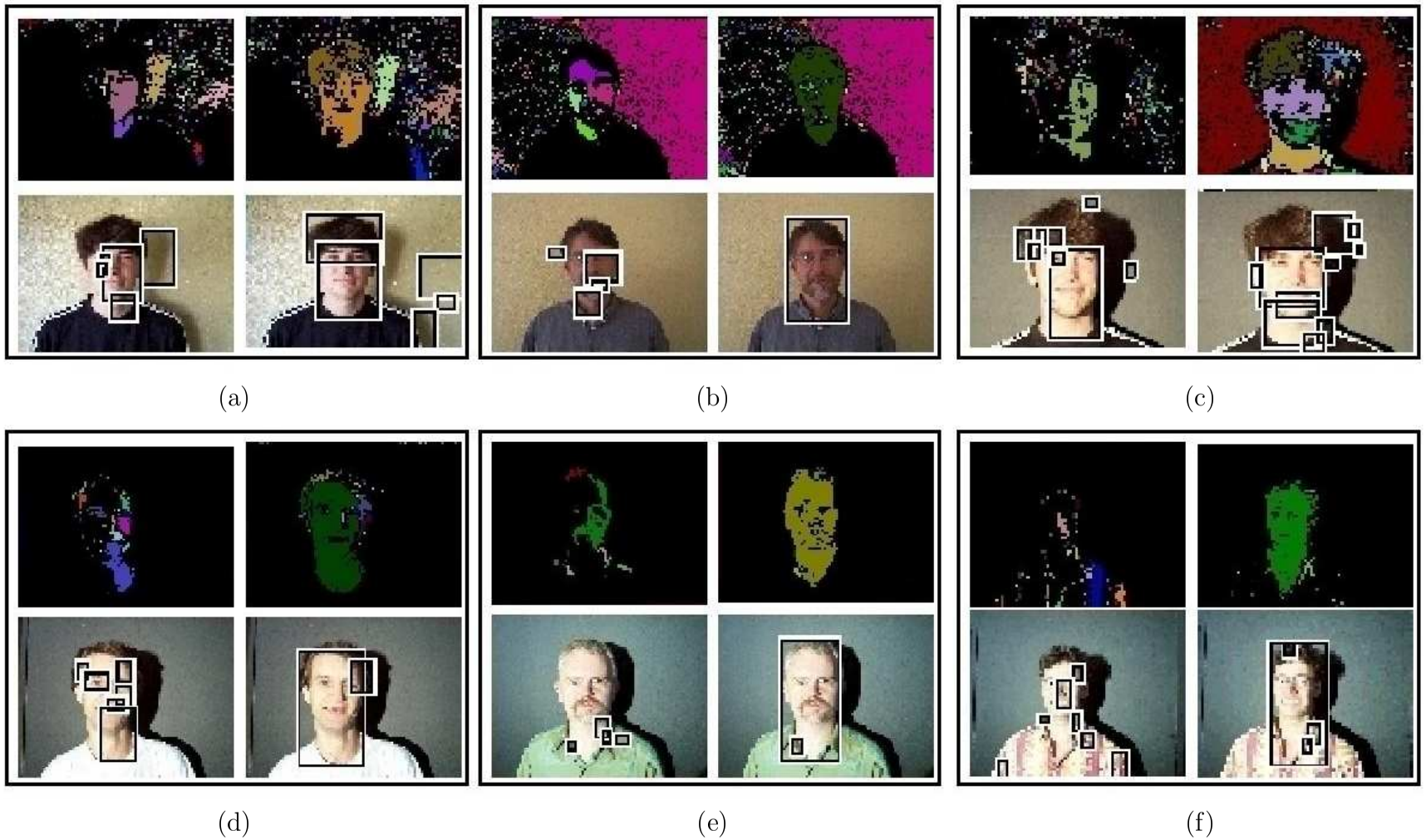


Figure 3.7. Nonlinear color transform. Six detection examples, with and without the transform are shown. For each example, the images shown in the first column are skin regions and detections without the transform, while those in the second column are results with the transform.

3.3 Localization of Facial Features

Among the various facial features, eyes, mouth, and face boundary are the most prominent features for recognition [103] and for estimation of 3D head pose [161], [162]. Most approaches for eye [163], [164], [165], [166], [167], mouth [165], [168], face boundary [165], and face [20] localization are template based. However, our approach is able to directly locate eyes, mouth, and face boundary based on their feature maps derived from the the luma and the chroma of an image, called the eye map, the mouth map and the face boundary map, respectively. For computing the eye map and the mouth map, we consider only the area covered by a *face mask* that is built by enclosing the grouped skin-tone regions with a pseudo convex hull, which is constructed by connecting the boundary points of skin-tone regions in horizontal and vertical directions. Figure 3.8 shows an example of the face mask.

(a) (b) (c) (d)

Figure 3.8. Construction of the face mask. (a) Face candidates; (b) one of the face candidates; (c) grouped skin areas; (d) the face mask.

3.3.1 Eye Map

We first build two separate eye maps, one from the chrominance components and the other from the luminance component of the color image. These two maps are then combined into a single eye map. The eye map from the chroma is based on the observation that high C_b and low C_r values are found around the eyes. It is constructed from information contained in C_b, the inverse (negative) of C_r, and the ratio C_b/C_r, as described in Eq. (3.1).

$$EyeMapC = \frac{1}{3}\{ (C_b^2) + (\tilde{C}_r)^2 + (C_b/C_r) \}, \tag{3.1}$$

where C_b^2, $(\tilde{C}_r)^2$, and C_b/C_r all are normalized to the range $[0, 255]$ and $\tilde{C}r$ is the negative of C_r (i.e., $255 - C_r$). An example of the eye map from the chroma is shown in Fig. 3.9(a).

The eyes usually contain both dark and bright pixels in the luma component. Based on this observation, grayscale morphological operators (e.g., dilation and erosion) [169] can be designed to emphasize brighter and darker pixels in the luma component around eye regions. These operations have been used to construct feature vectors for face images at multiple scales for frontal face authentication [66]. We use grayscale dilation and erosion with a hemispheric structuring element at a single estimated scale to construct the eye map from the luma, as described in Eq. (3.2).

$$EyeMapL = \frac{Y(x,y) \oplus g_\sigma(x,y)}{Y(x,y) \ominus g_\sigma(x,y) + 1} \quad , \tag{3.2}$$

where the grayscale dilation $\oplus$ and erosion $\ominus$ operations [169] on a function $f : \mathcal{F} \subset R^2 \longrightarrow R$ using a structuring function $g : \mathcal{G} \subset R^2 \longrightarrow R$ are defined as follows.

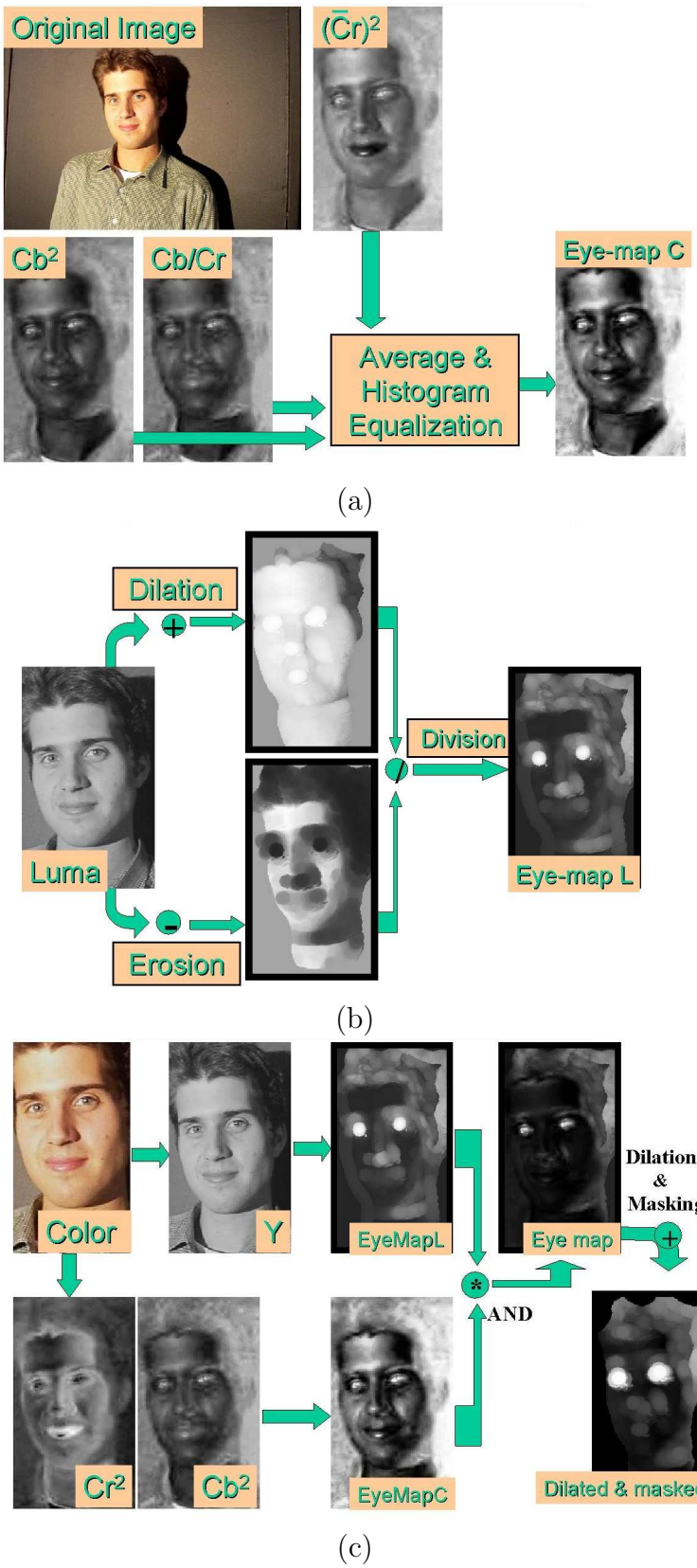


Figure 3.9. Construction of eye maps: (a) from chroma; (b) from luma; (c) the combined eye map.

$$(f \oplus g_\sigma)(x,y) = \mathbf{Max}\{f(x-c, y-r) + g(c,r)\}; \\ (x-c, y-r) \in \mathcal{F}, \quad (c,r) \in \mathcal{G}\ , \tag{3.3}$$

$$(f \ominus g_\sigma)(x,y) = \mathbf{Min}\{f(x-c, y-r) + g(c,r)\}; \\ (x-c, y-r) \in \mathcal{F}, \quad (c,r) \in \mathcal{G}\ , \tag{3.4}$$

$$g_\sigma(x,y) = \begin{cases} |\sigma| \cdot \left(\left| 1 - (R(x,y)/\sigma)^2 \right|^{1/2} - 1 \right); & R \le |\sigma|, \\ -\infty; & R > |\sigma|, \end{cases} \tag{3.5}$$

$$R(x,y) = \sqrt{x^2 + y^2}, \tag{3.6}$$

where σ is a scale parameter, which will be described later in Eq. (3.11). An example of a hemispheric structuring element is shown in Fig. 3.10. The construction of the

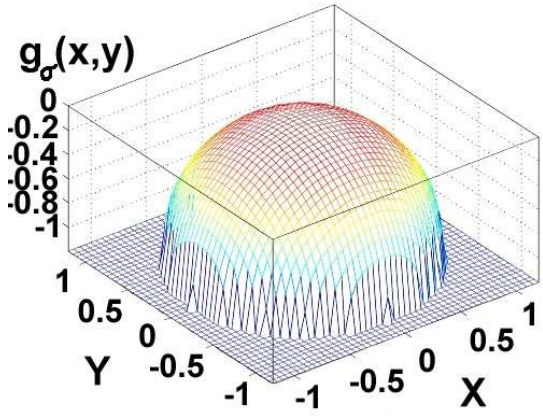


Figure 3.10. An example of a hemispheric structuring element for grayscale morphological dilation and erosion with $\sigma = 1$.

eye map from the luma is illustrated in Fig. 3.9(b). Note that before performing the grayscale dilation and erosion operations, we fill the background of the face mask with the mean value of the luma in the face mask (skin regions) in order to smooth the noisy boundary of detected skin areas.

The eye map from the chroma is enhanced by histogram equalization, and then combined with the eye map from the luma by an AND (multiplication) operation in Eq. (3.7).

$$EyeMap = (\ EyeMapC\ )\quad \textbf{AND}\quad (\ EyeMapL\ )\ . \tag{3.7}$$

The resulting eye map is dilated, masked, and normalized to brighten the eyes and suppress other facial areas, as can be seen in Fig. 3.9(c). The locations of the eye candidates are initially estimated from the pyramid decomposition of the eye map, and then refined using iterative thresholding and binary morphological closing on this eye map.

3.3.2 Mouth Map

The color of mouth region contains more red component compared to the blue component than other facial regions. Hence, the chrominance component C_r, proportional to $(red - Y)$, is greater than C_b, proportional to $(blue - Y)$, near the mouth areas. We further notice that the mouth has a relatively low response in the C_r/C_b feature, but it has a high response in C_r^2. We construct the mouth map as follows:

$$MouthMap \;=\; C_r^2 \cdot \left(C_r^2 - \eta \cdot C_r/C_b\right)^2 \;; \tag{3.8}$$

$$\eta \;=\; 0.95 \cdot \frac{\frac{1}{n}\sum\limits_{(x,y)\in\mathcal{FG}} C_r(x,y)^2}{\frac{1}{n}\sum\limits_{(x,y)\in\mathcal{FG}} C_r(x,y)/C_b(x,y)}\ , \tag{3.9}$$

where both C_r^2 and C_r/C_b are normalized to the range $[0, 255]$, and n is the number of pixels within the face mask, $\mathcal{FG}$. The parameter η is estimated as the ratio of the

average C_r^2 to the average C_r/C_b. Figure 3.11 shows the major steps in computing the mouth map of the subject in Fig. 3.9. Note that after the mouth map is dilated, masked, and normalized, it is dramatically brighter near the mouth areas than at other facial areas.

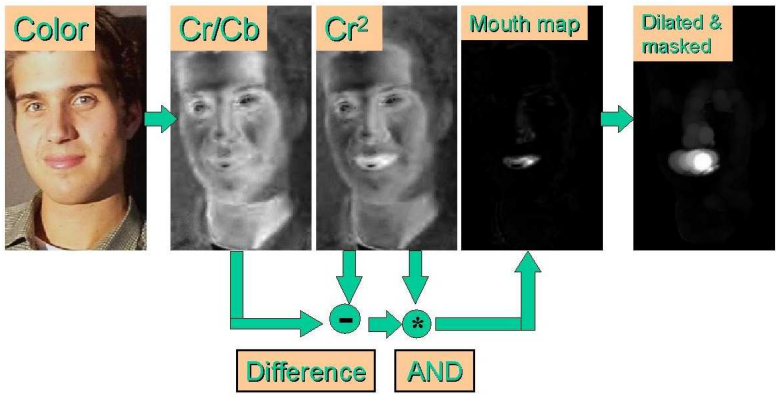


Figure 3.11. Construction of the mouth map.

3.3.3 Eye and Mouth Candidates

We form an *eye-mouth triangle* for all possible combinations of two eye candidates and one mouth candidate within a face candidate. We then verify each eye-mouth triangle by checking (i) luma variations and average gradient orientations of eye and mouth blobs; (ii) geometry and orientation constraints of the triangle; and (iii) the presence of a face boundary around the triangle. A weight is computed for each verified eye-mouth triangle. The triangle with the highest weight that exceeds a threshold is selected. We discuss the detection of face boundary in Section 3.3.4, and the selection of the weight and the threshold in Section 3.3.5.

Note that the eye and mouth maps are computed within the entire areas of the face candidate, which is bounded by a rectangle. The search for the eyes and the

mouth is performed within the face mask. The eye and mouth candidates are located by using (i) a pyramid decomposition of the eye/mouth maps and (ii) an iterative thresholding and binary morphological closing on the enhanced eye and mouth maps. The number of pyramid levels, L, is computed from the size of the face candidate, as defined in Eqs. (3.10) and (3.11).

$$L = \mathbf{Max}\{\ \lceil \log_2(2\sigma) \rceil, \lfloor \log_2(\mathbf{Min}(W, H)/F_c) \rfloor\ \}\ ; \tag{3.10}$$

$$\sigma = \lfloor \sqrt{W \cdot H}\ /\ (2 \cdot F_e) \rfloor\ , \tag{3.11}$$

where W and H represent the width and height of the face candidate; $F_c \times F_c$ is the minimum expected size of a face candidate; σ is a spread factor selected to prevent the algorithm from removing small eyes and mouths in the morphological operations; and F_e is the maximal ratio of an average face size to the average eye size. In our implementation, F_c is 7 pixels, and F_e is 12 pixels.

The coarse locations of eye and mouth candidates obtained from the pyramid decomposition are refined by checking the existence of eyes/mouth blobs which are obtained after iteratively thresholding and (morphologically) closing the eye and mouth maps. The iterative thresholding starts with an initial threshold value, reduces the threshold step by step, and stops when either the threshold falls below a stopping value or when the number of feature candidates reaches pre-determined upper bounds, N_{eye} for the eyes and N_{mth} for the mouth. The threshold values are automatically computed as follows.

$$Th = \frac{\alpha}{n} \sum_{(x,y) \in \mathcal{FG}} Map(x, y) + (1 - \alpha) \cdot \underset{(x,y) \in \mathcal{FG}}{\mathbf{Max}} Map(x, y), \tag{3.12}$$

where $Map(x, y)$ is either the eye or the mouth map; the parameter α is equal to 0.5 for the initial threshold value, and is equal to 0.8 for the stopping threshold. The use of upper bounds on the number of eye and mouth candidates can prevent the algorithm from spending too much time in searching for facial features. In our implementation, the maximum number of eye candidates, N_{eye}, is 8 and the maximum number of mouth candidates, N_{mth}, is 5.

3.3.4 Face Boundary Map

Based on the locations of eyes/mouth candidates, our algorithm first verifies whether the average orientation of luma gradients around each eye matches the interocular direction, and then constructs a face boundary map from the luma. Finally, it utilizes the Hough transform to extract the best-fitting ellipse. The fitted ellipse is used for computing the eye-mouth triangle weight. Figure 3.12 shows the boundary map that is constructed from both the magnitude and the orientation components of the luma gradient within the regions that have positive orientations of the gradient orientations (i.e., have counterclock-wise gradient orientations). We have modified Canny edge detection [170] algorithm to compute the gradient of the luma as follows. The gradient of a luma subimage, $S(x, y)$, which is slightly larger than the face candidate in size is estimated by

$$\nabla S(x, y) = (G_x, G_y) = (D_\sigma(x) \circledast S(x, y), D_\sigma(y) \circledast S(x, y)), \quad (3.13)$$

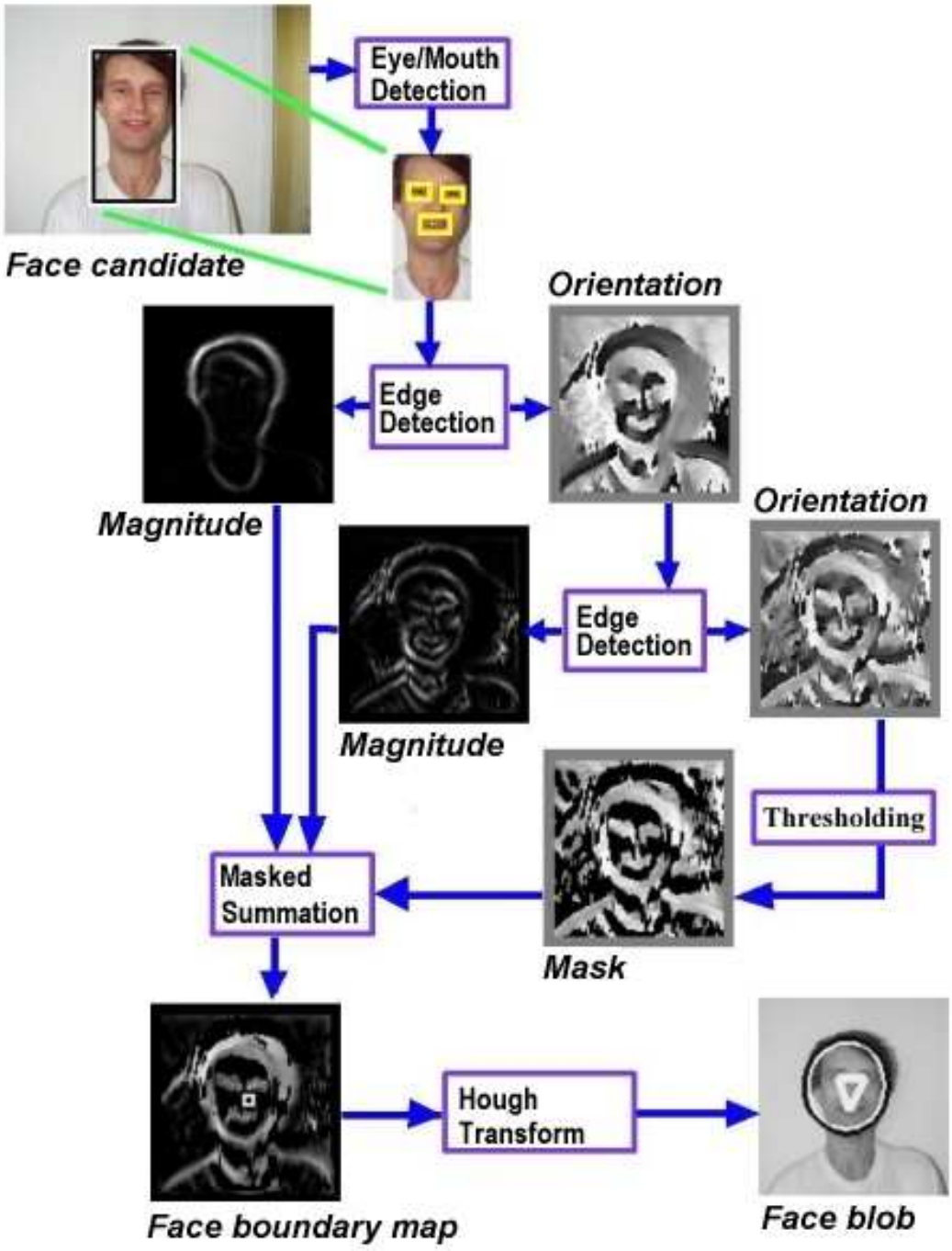


Figure 3.12. Computation of face boundary and the eye-mouth triangle.

where $D_\sigma(x)$ is the derivative of the Gaussian with zero mean and variance σ^2, and $\circledast$ is the convolution operator. Unlike the Canny edge detector, our edge detection requires only a single standard deviation σ (a spread factor) for the Gaussian that is estimated from the size of the eye-mouth triangle.

$$\sigma = \left(\frac{-ws^2}{8 \ln(wh)} \right)^{1/2}; \qquad ws = \mathbf{Max}(dist_{io}, dist_{em}), \tag{3.14}$$

where ws is the window size for a Gaussian, which is the maximum value of the interocular distance ($dist_{io}$) and the distance between the interocular midpoint and the mouth ($dist_{em}$); $wh = 0.1$ is the desired value of the Gaussian distribution at the border of the window. In Fig. 3.12, the magnitudes and orientations of all gradients have been squared and scaled between 0 and 255. Fig. 3.12 shows that the gradient orientation provides more information to detect face boundaries than the gradient magnitude. So, an edge detection algorithm is applied to the gradient orientation and the resulting edge map is thresholded to obtain a mask for computing the face boundary. The gradient magnitude and the magnitude of the gradient orientation are masked, added, and scaled into the interval $[0, 1]$ to construct the face boundary map. The center of a face, indicated as a white rectangle in the face boundary map in Fig. 3.12, is estimated from the first-order moment of the face boundary map.

The Hough transform is used to fit an elliptical shape to the face boundary map. An ellipse in a plane has five parameters: an orientation angle, two coordinates of the

center, and lengths of major and minor axes. Since we know the locations of eyes and mouth, the orientation of the ellipse can be estimated from the direction of a vector that starts from the midpoint between the eyes towards the mouth. The location of the ellipse center is estimated from the face boundary map. Hence, we need only a two-dimensional accumulator for estimating the ellipse for bounding the face. The accumulator is updated by perturbing the estimated center by a few pixels for a more accurate localization of the ellipse.

3.3.5 Weight Selection for a Face Candidate

For each face in the image, our algorithm can detect several eye-mouth-triangle candidates that are constructed from eye and mouth candidates. Each candidate is assigned a weight which is computed from the eye and mouth maps, the maximum accumulator count in the Hough transform for ellipse fitting, and face orientation that favors vertical faces and symmetric facial geometry, as described in Eqs. (3.15)-(3.19). The eye-mouth triangle with the highest weight (face score) that is above a threshold is retained. In Eq. (3.15), the triangle weight, $tw(i, j, k)$, for the i-th and the j-th eye candidates and the k-th mouth candidate is the product of the eye-mouth weight, $emw(i, j, k)$, the face-orientation weight, $ow(i, j, k)$, and boundary quality, $q(i, j, k)$. The eye-mouth weight is the average of the eye-pair weight, $ew(i, j)$, and the mouth weight, $mw(k)$, as described in Eq. (3.16).

$$tw(i,j,k) = emw(i,j,k) \cdot ow(i,j,k) \cdot q(i,j,k); \tag{3.15}$$

$$emw(i,j,k) = \frac{1}{2}(ew(i,j) + mw(k)); \tag{3.16}$$

$$ew(i,j) = \frac{EyeMap(x_i,y_i) + EyeMap(x_j,y_j)}{2 \cdot EyeMap(x_m,y_m)}; \quad i > j; \;\; i,j \in [1, N_{eye}]; \tag{3.17}$$

$$mw(k) = \frac{MouthMap(x_k,y_k)}{MouthMap(x_m,y_m)}; \qquad k \in [1, N_{mth}]; \tag{3.18}$$

$$ow(i,j,k) = \prod_{r=1}^{2} e^{-3(1-cos^2(\theta_r(i,j,k)))}; \quad cos(\theta_r(i,j,k)) = \frac{\vec{u_r} \cdot \vec{v_r}}{\|\vec{v_r}\|}; \quad \|\vec{v_r}\| = 1. \tag{3.19}$$

Eq. (3.17) describes the eye-pair weight which is the normalized average of the eye map value around the two eyes, where $EyeMap(x_i, y_i)$ is the eye map value for the i-th eye candidate (associated with an eye blob and a corresponding pixel in the lowest level of the image pyramid). $EyeMap(x_m, y_m)$ is the eye map value for the most significant eye candidate (having the highest response within the eye map). The mouth weight, $mw(k)$ in Eq. (3.18), is obtained by normalizing the mouth map value at the k-th mouth candidate (i.e., a mouth blob), $MouthMap(x_k, y_k)$, by the mouth map value at the most significant mouth candidate, $MouthMap(x_m, y_m)$. The face-orientation weight, described in Eq. (3.19), is the product of two attenuation terms, each of which is an exponential function of a projection ($cos\theta_r$) of a vector ($\vec{v_r}$) along a particular direction ($\vec{u_r}$), where $r = 1, 2$. As can be seen in Fig. 3.13, one term favors a symmetric face, and it is a projection ($cos\theta_1$) of the vector $\vec{v_1}$ (from the midpoint of the two eyes to the mouth) along a vector ($\vec{u_1}$) that is perpendicular to the interocular

segment. The other term favors an upright face, and is a projection of a vector $\vec{v_2}$ (from the mouth to the midpoint of the two eyes) along the vertical axis ($\vec{u_2}$) of the image plane. The exponential function, shown in Fig. 3.14, is designed such that the attenuation has the maximal value of 1 when $\theta_1 = \theta_2 = 0°$ (i.e., when eyes and mouth form a letter "T" or equivalently the face is upright), and it decreases to below 0.5 at $\theta_1 = \theta_2 = 25°$. The quality of face boundary, $q(i, j, k)$, can be directly obtained from the votes received by the best elliptical face boundary in the Hough transform.

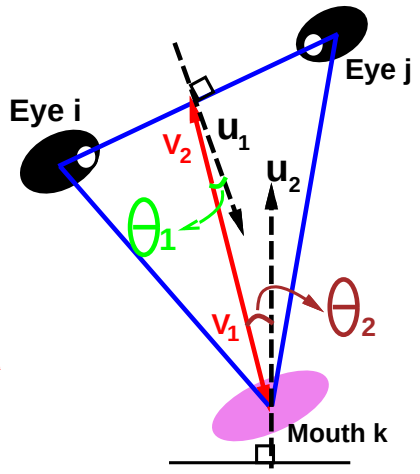


Figure 3.13. Geometry of an eye-mouth triangle, where $\vec{v_1} = -\vec{v_2}$; unit vectors $\vec{u_1}$ and $\vec{u_2}$ are perpendicular to the interocular segment and the horizontal axis, respectively.

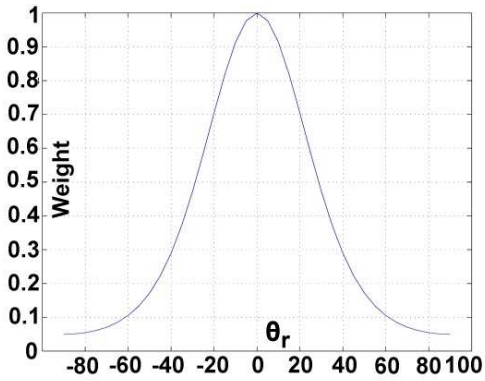


Figure 3.14. Attenuation term, $e^{-3(1-cos^2(\theta_r(i,j,k)))}$, plotted as a function of the angle θ_r (in degrees) has a maximal value of 1 at $\theta_r = 0°$, and a value of 0.5 at $\theta_r = 25°$.

The pose-oriented threshold for the face score is empirically determined and used for removing false positives (0.16 for near-frontal views and 0.13 for half-profile views). The face pose (frontal vs. profile) is estimated by comparing the distances from each of the two eyes to the major axis of the fitted ellipse.

3.4 Experimental Results

We have evaluated our algorithm on several face image databases, including family and news photo collections. Face databases designed for face recognition, including the FERET face database [28], usually contain grayscale mugshot-style images, therefore, in our opinion, are not suitable for evaluating face detection algorithms. Most of the commonly used databases for face detection, including the Carnegie Mellon University (CMU) database, contain grayscale images only. Therefore, we have constructed our databases for face detection from MPEG7 videos, the World Wide Web, and personal photo collections. These color images have been taken taken under varying lighting conditions and with complex backgrounds. Further, these images have substantial variability in quality and they contain multiple faces with variations in color, position, scale, orientation, 3D pose, and facial expression.

Our algorithm can detect multiple faces of different sizes with a wide range of facial variations in an image. Further, the algorithm can detect both dark skin-tones and bright skin-tones because of the nonlinear transformation of the $C_b - C_r$ color space. All the algorithmic parameters in our face detector have been empirically determined; same parameter values have been used for all the test images. Figure

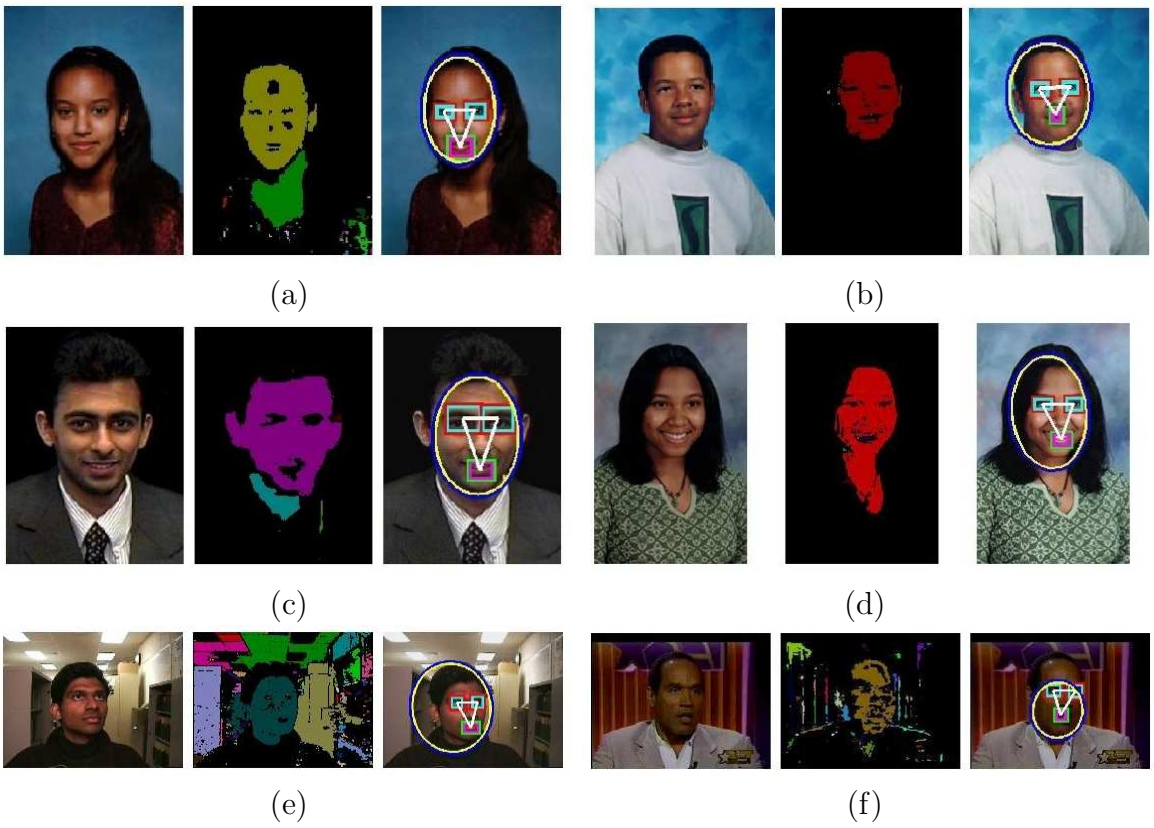
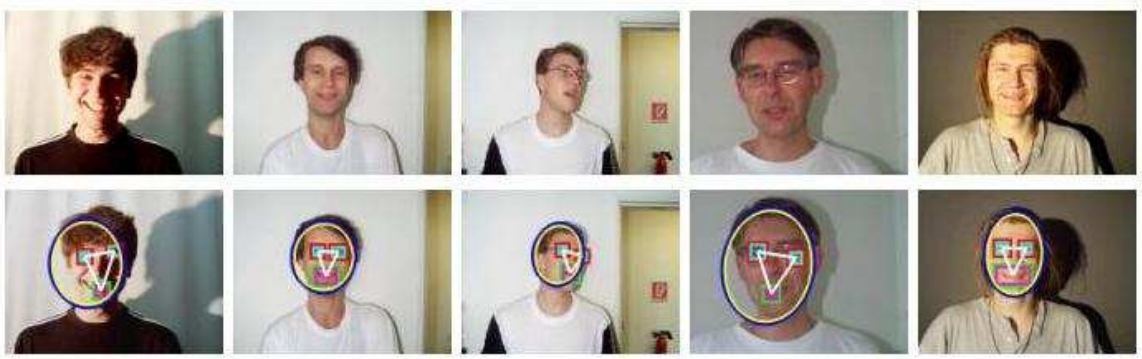


Figure 3.15. Face detection examples containing dark skin-tone faces. Each example contains an input image, grouped skin regions shown in pseudo color, and a lighting-compensated image overlaid with detected face and facial features.

3.15 demonstrates that our algorithm can successfully detect dark skin faces. Figure 3.16 shows the results for subjects with some facial variations (e.g., closed eyes or open mouth). Figure 3.17 shows detected faces for subjects who are wearing glasses. The eye glasses can break up the detected skin tone components of a face into smaller components, and cause reflections around the eyes. Figure 3.18 shows that the proposed algorithm is not sensitive to the presence of facial hair (moustache and beard). Figure 3.19 demonstrates that our algorithm can detect non-frontal faces as long as the eyes and mouth are visible in half-profile views.

A summary of the detection results (including the number of false positives, de-



(a) (b) (c) (d) (e)

Figure 3.16. Face detection results on closed-eye or open-mouth faces. Each example contains an original image (top) and a lighting-compensated image (bottom) overlaid with face detection results.

(a) (b) (c) (d) (e)

Figure 3.17. Face detection results in the presence of eye glasses. Each example contains an original image (top) and a lighting-compensated image (bottom) overlaid with face detection results.

tection rates, and average CPU time for processing an image) on the HHI MPEG7 image database [15] and the Champion database [171] are presented in Tables 3.1 and 3.2, respectively. Note that the detection rate depends on the database. The HHI image database contains 206 images, each of size 640×480 pixels. Subjects in the HHI image database belong to several racial groups. Lighting conditions (including overhead lights and side lights) change from one image to another. Further, these images contain frontal, near-frontal, half-profile, and profile face views of different sizes. A detected face is a *correct* detection if the detected locations of the eyes,

(a) (b) (c) (d) (e)

Figure 3.18. Face detection results for subjects with facial hair. Each example contains an original image (top) and a lighting-compensated image (bottom) overlaid with face detection results.

the mouth, and the ellipse bounding a human face are found with a small amount of tolerance, otherwise it is called a *false positive*. The detection rate is computed by the ratio of the number of correct detections in a gallery to that of all human faces in the gallery. Figure 3.20(a) shows a subset of the HHI images. The detection results of our algorithm are shown in three stages. In the first stage, we show the skin-tone regions (Fig. 3.20(b)) using pseudo-color; different colors correspond to different skin-tone groups. In the second stage, we fuse bounding rectangles that have significant overlapping areas with neighboring rectangles (Fig. 3.20(c)). Each bounding rectangle indicates a face candidate. In the third stage, we locally detect facial features for each face candidate. Figure 3.20(d) shows the final detection results after these three stages. The detected faces are depicted by yellow-blue ellipses, and the detected facial features (eyes and mouth) are connected by a triangle. The detection rates and the number of false positives for different poses are summarized in Table 3.1. The detection rate after the first two stages is about 97% for all poses. After the third stage, the detection rate decreases to 89.40% for frontal faces, and to

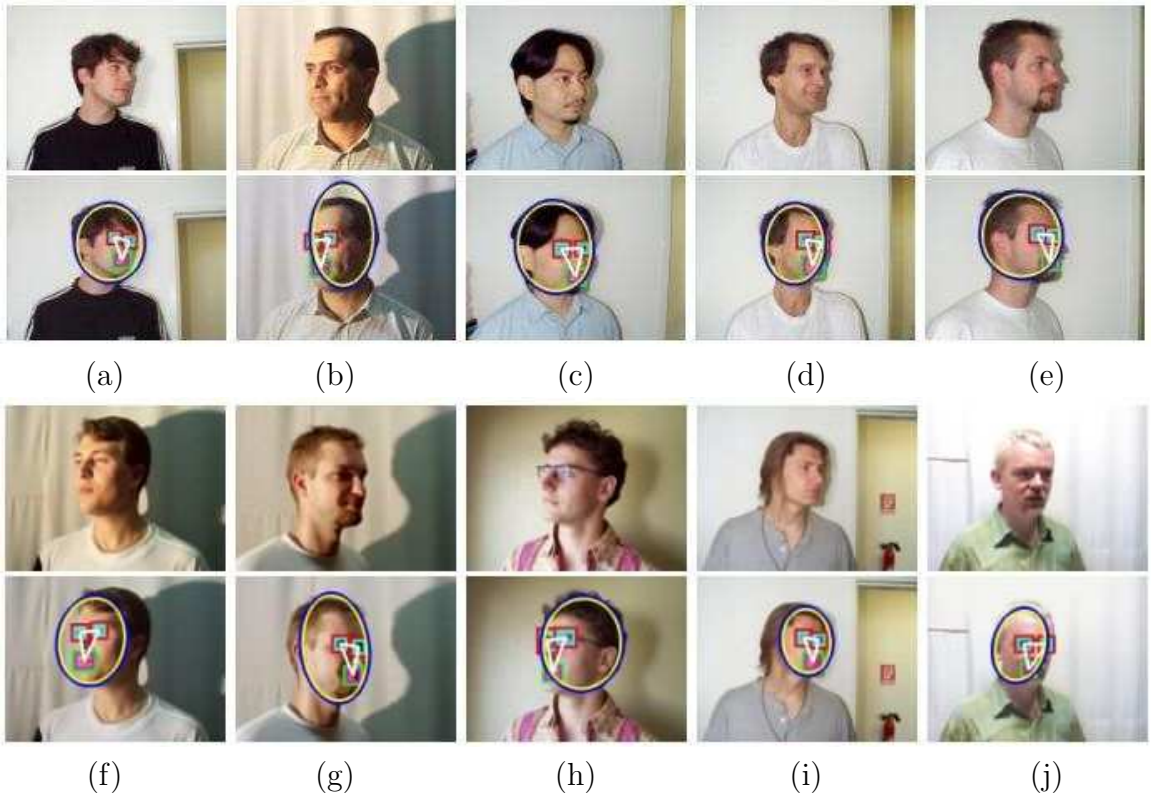


Figure 3.19. Face detection results on half-profile faces. Each example contains an original image (top) and a lighting-compensated image (bottom) overlaid with face detection results.

90.74% for near-frontal faces, and to 74.67% for half-profile faces. The reason for this decrease in detection rate is the removal of those faces in which the eyes/mouth are not visible. However, we can see that the number of false positives is dramatically reduced from $9,406$ after the skin grouping stage to just 27 after the feature detection stage for the whole database containing 206 images.

The Champion database was collected from the Internet, and contains 227 *compressed* images which are approximately 150×220 pixels in size. Because most of the images in this database are captured in frontal and near-frontal views, we present a single detection rate for all poses in Table 3.2. The detection rate for the first two stages is about 99.12%. After the third stage, the detection rate decreases to 91.63%.

The number of false positives is also dramatically reduced from $5,582$ to 14. We present face detection results on a subset of the Champion database in Fig. 3.21. Figure 3.22 shows the detection results on a collection of family photos (total of 55 images). Figure 3.23 shows results on a subset of news photos (total of 327 images) downloaded from the Yahoo news site [172]. As expected, detecting faces in family group and news pictures is more challenging, but our algorithm is able to perform quite well on these images. Detection rate on the collection of 382 family and news photos (1.79 faces per image) is 80.35%, and the false positive rate (the ratio of the number of false positives to the number of true faces) is 10.41%. More results are available at http://www.cse.msu.edu/∼hsureinl/facloc/.

Table 3.1
Detection results on the HHI image database (Image size 640×480) on a PC with 1.7 GHz CPU. FP: False Positives, DR: Detection Rate.

Head Pose	Frontal	Near-Frontal	Half-Profile	Profile	Total
No. of images	66	54	75	11	206
Stage 1: Grouped skin regions					
No. of FP	3145	2203	3781	277	9406
DR (%)	95.45	98.15	96.00	100	96.60
Time (sec): average $\pm$ s. d.	1.56 ± 0.45				
Stage 2: Rectangle merge					
No. of FP	468	287	582	39	1376
DR (%)	95.45	98.15	96.00	100	96.60
Time (sec): average $\pm$ s. d.	0.18 ± 0.23				
Stage 3: Facial feature detection					
No. of FP	4	6	14	3	27
DR (%)	89.40	90.74	74.67	18.18	80.58
Time (sec): average $\pm$ s. d.	22.97 ± 17.35				

Table 3.2
Detection results on the Champion database (Image size $\sim 150 \times 220$) on a PC with 860 MHz CPU. FP: False Positives, DR: Detection Rate.

Stage	1	2	3
No. of images	227		
No. of FP	5582	382	14
DR (%)	99.12	99.12	91.63
Time (sec): average $\pm$ s. d.	0.080 ± 0.036	0.012 ± 0.020	5.780 ± 4.980

(a) (b) (c) (d)

Figure 3.20. Face detection results on a subset of the HHI database: (a) input images; (b) grouped skin regions; (c) face candidates; (d) detected faces are overlaid on the lighting-compensated images.

(a) (b) (c) (d)

Figure 3.21. Face detection results on a subset of the Champion database: (a) input images; (b) grouped skin regions; (c) face candidates; (d) detected faces are overlaid on the lighting-compensated images.

Figure 3.22. Face detection results on a subset of eleven family photos. Each image contains multiple human faces. The detected faces are overlaid on the color-compensated images. False negatives are due to extreme lighting conditions and shadows. Notice the difference between the input and color-compensated images in terms of color balance. The bias color in the original images has been compensated in the resultant images.

Figure 3.22. (Cont'd).

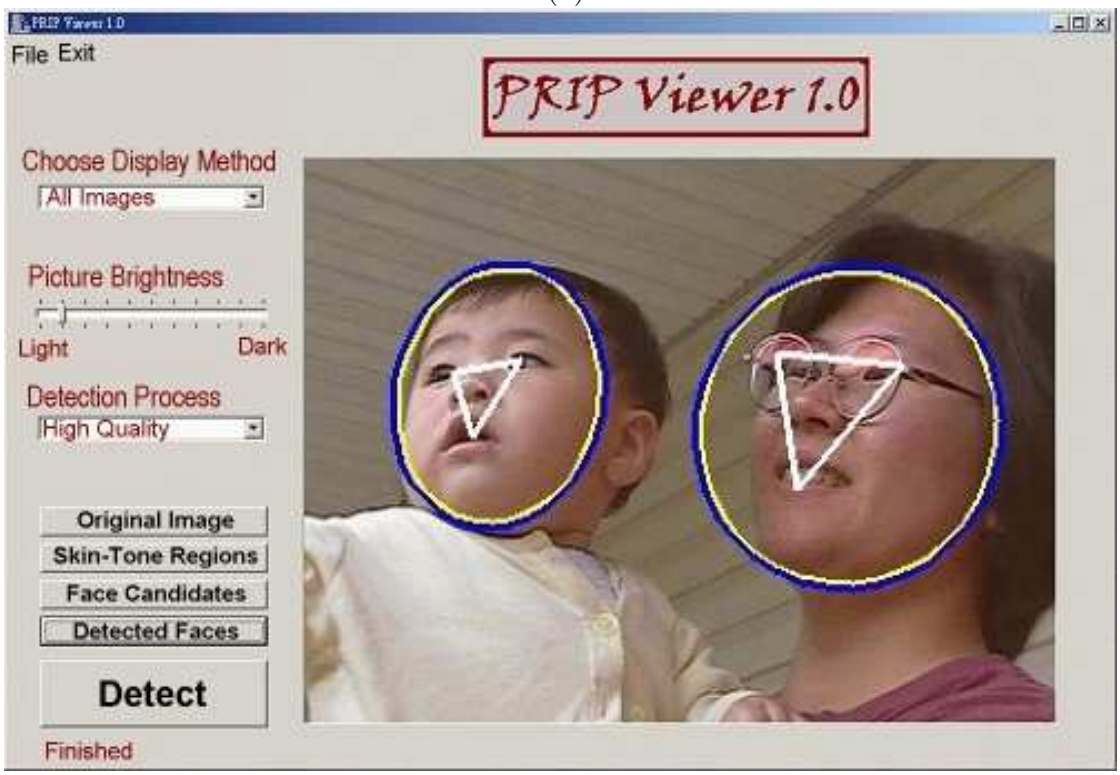
Figure 3.22. (Cont'd).

Figure 3.23. Face detection results on a subset of 24 news photos. The detected faces are overlaid on the color-compensated images. False negatives are due to extreme lighting conditions, shadows, and low image quality (i.e., high compression rate).

3.5 Summary

We have presented a face detection algorithm for color images using a skin-tone color model and facial features. Our method first corrects the color bias by a lighting compensation technique that automatically estimates the reference white pixels. We overcome the difficulty of detecting the low-luma and high-luma skin tones by applying a nonlinear transform to the YC_bC_r color space. Our method detects skin regions over the entire image, and then generates face candidates based on the spatial arrangement of these skin patches. It then constructs eye, mouth, and boundary maps for detecting the eyes, mouth, and face boundary, respectively. The face candidates are further verified by the presence of these facial features. Detection results on several photo collections have been demonstrated. Our goal is to design a system that detects faces and facial features, allows users to edit detected faces (via the user interface shown in Fig. 3.24), and uses these detected facial features as indices for identification and for retrieval from image and video databases.

(a)



(b)

Figure 3.24. Graphical user interface (GUI) for face editing: (a) detection mode; (b) editing mode.

Chapter 4

Face Modeling

We first introduce an overview of our modeling method [173], and describe the generic face model and facial measurements. Then we present an approach for adapting the generic model to the facial measurements. Finally, an adapted 3D face model of an individual is texture-mapped and reproduced at different viewpoints for visualization and recognition.

4.1 Modeling Method

For efficiency, we construct a 3D model of a human face from *a priori* knowledge (a generic face model) of the geometry of the human face. The generic face model is a triangular mesh, whose vertices can precisely specify facial features that are crucial for recognition, such as eyebrows, eyes, nose, mouth, and face boundary. We call these features recognition-oriented features. The locations and associated properties of these recognition-oriented features are extracted from color texture and range data (or

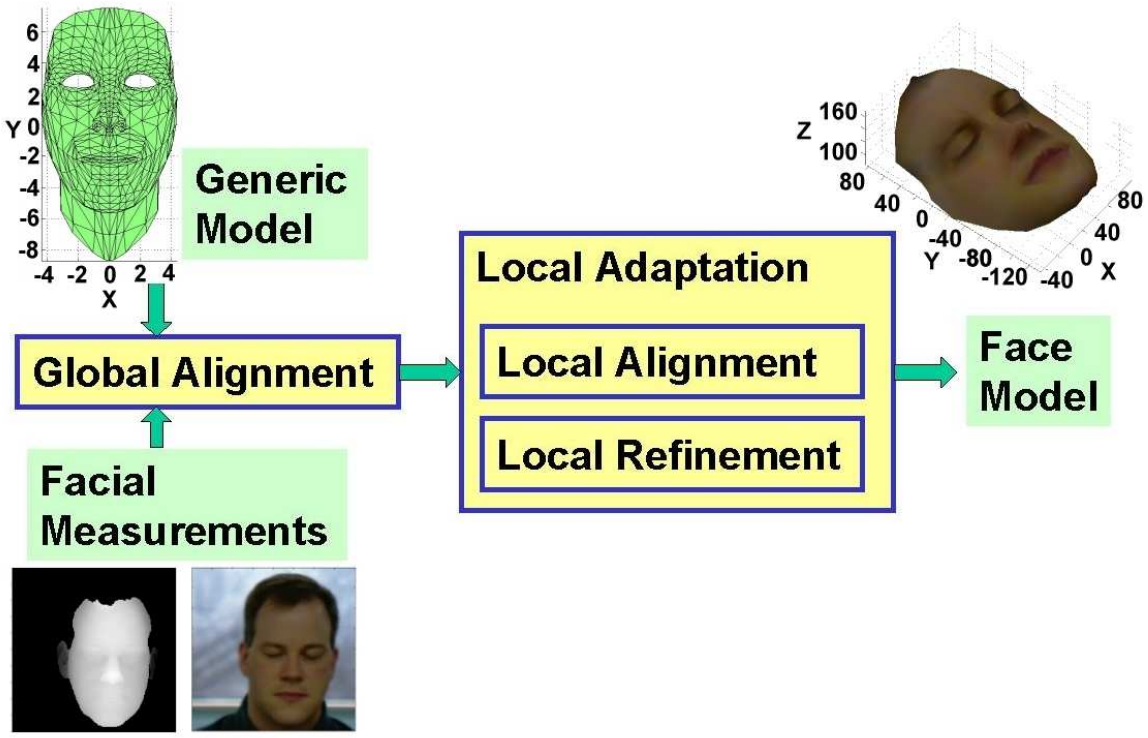


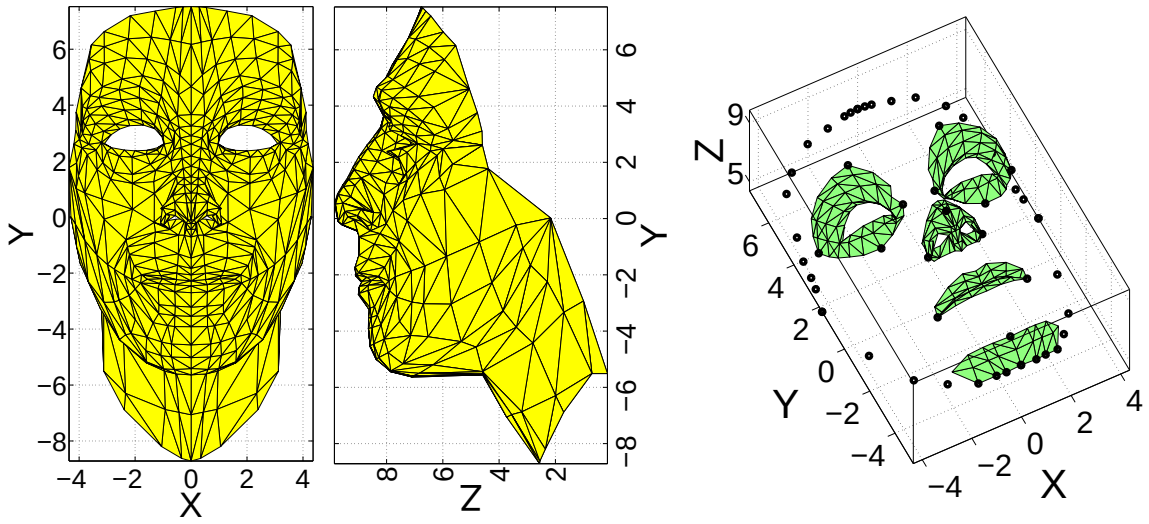
Figure 4.1. The system overview of the proposed modeling method based on a 3D generic face model.

disparity maps) obtained for an individual. The generic face model is modified so that these recognition-oriented features are fitted to the individual's facial geometry. The modeling process aligns and adapts the generic face model to the facial measurements in a global-to-local fashion. The overview of our face modeling method is given in Fig. 4.1. The input to the modeling algorithm is the generic face model and the facial measurements. The modeling method contains two major modules: (i) global alignment and (ii) local adaptation. The global alignment module changes the size of the generic face model, and aligns the scaled generic model according to the 3D head pose. The local adaptation module refines the facial features of the globally aligned generic face model iteratively and locally. We do not extract isosurfaces directly from facial measurements because facial measurements are often noisy (e.g., near the ears

and nose in frontal views), and because the extraction is time-consuming and usually generates triangles of the same size in the mesh. Hence, in our model construction, the desired recognition-oriented facial features can be specified and gradually modified in the 3D generic face model. The modeling algorithm generates an adapted/learned 3D face model with aligned facial texture. The 2D projections of the texture-mapped 3D model are further used for face verification and recognition.

4.2 Generic Face Model

We choose Waters' animation model [69], which contains 256 vertices and 441 facets for one half of the face, because this model captures most of the facial features that are needed for face recognition (as well as animation), and because triangular meshes are suitable for free-form surfaces like faces [136]. Figure 4.2 shows the frontal and one side view of the model, and facial features such as eyes, nose, mouth, face boundary, and chin. There are openings at both the eyes and the mouth, which can be manipulated. The Phong-shaded appearance of this model is shown for three different views in Fig. 4.3.



(a) (b) (c)

Figure 4.2. 3D triangular-mesh model and its feature components: (a) the frontal view; (b) a side view; (c) feature components.

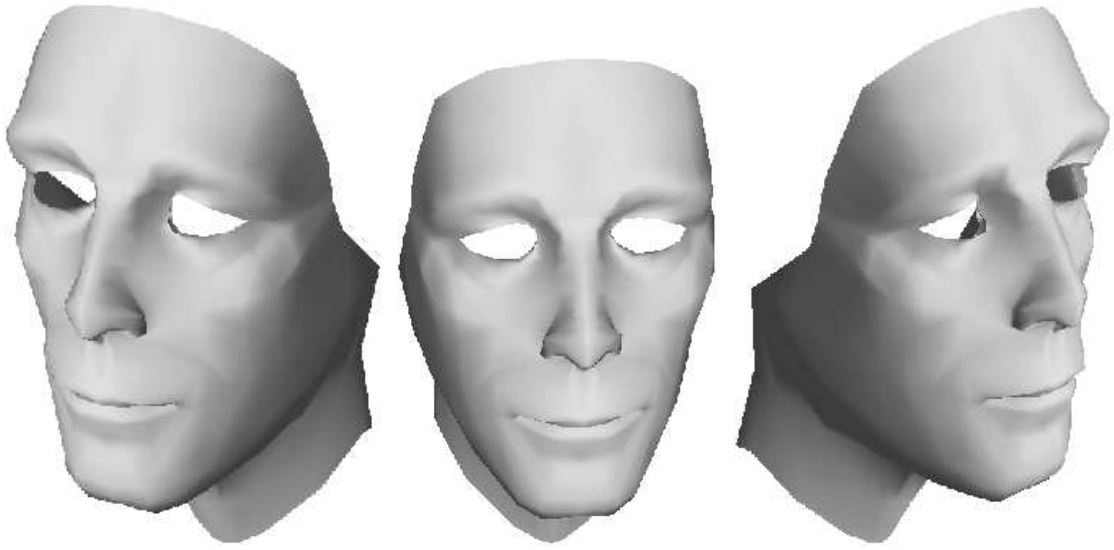
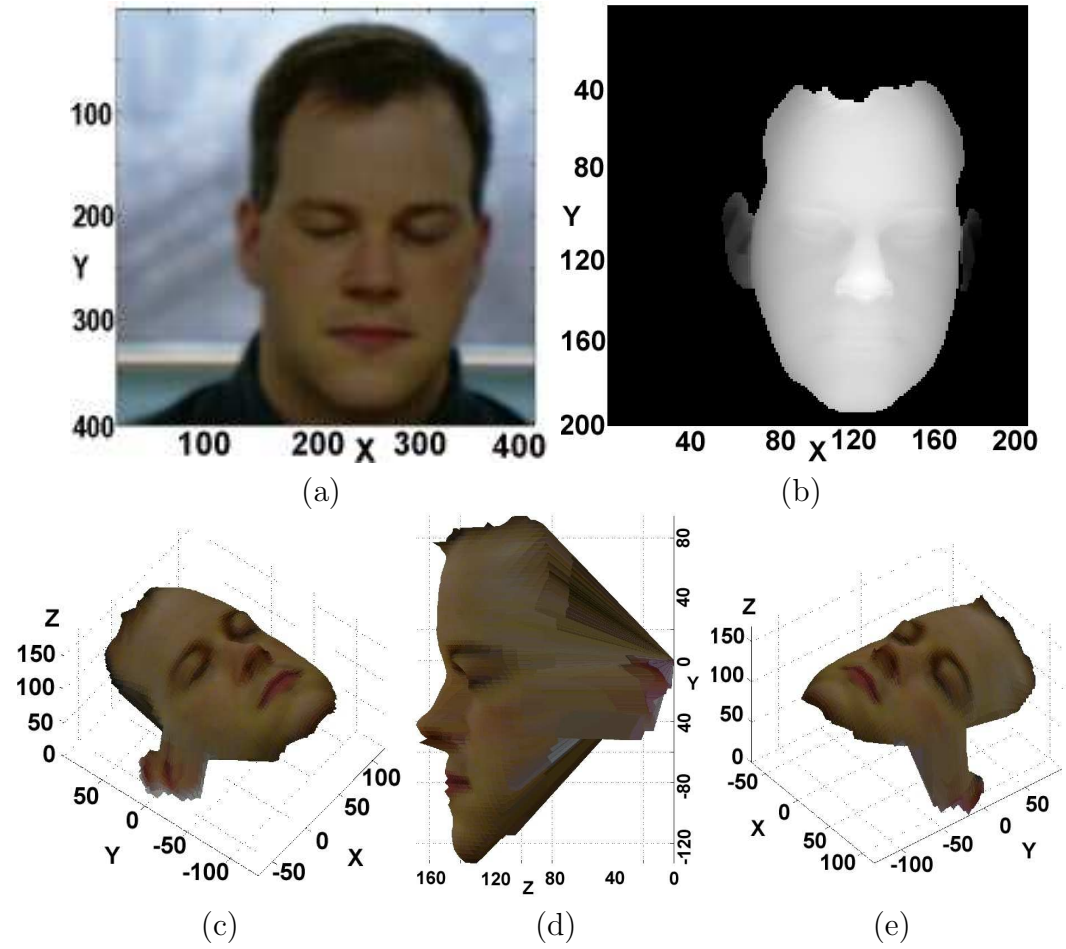


Figure 4.3. Phong-shaded 3D model shown at three viewpoints. Illumination is in front of the face model.

4.3 Facial Measurements

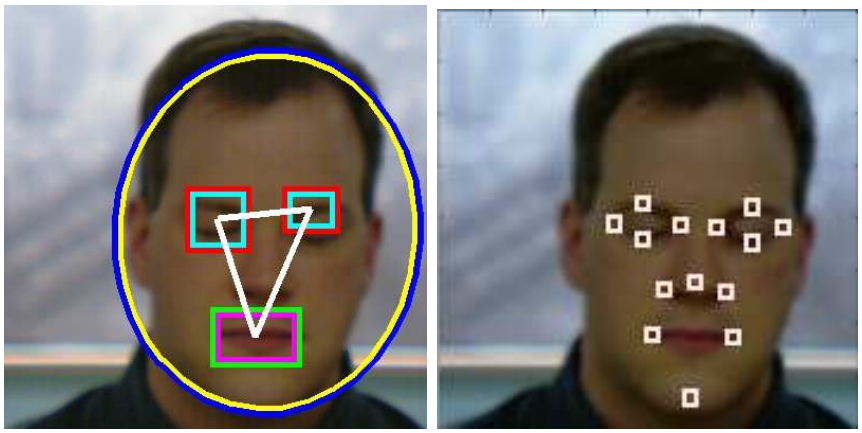
Facial measurements should include information about face shape and facial texture. 3D shape information can be derived from a stereo pair, a collection of frames in a video sequence, or shape from shading. It can also be obtained directly from range data. We use the range database of human faces [174], which was acquired using a Minolta Vivid 700 digitizer. The digitizer generates a registered 200×200 range map and a 400×400 color image for each acquisition. Figure 4.4 shows a color image and a range map of a frontal view, and the texture-mapped appearance from three different views. The locations of face and facial features such as eyes and mouth in the color texture image can be detected by the face detection algorithm described in Chapter 3 [175] (see Fig. 4.5(a)). The corners of eyes, mouth, and nose can be easily obtained based on the locations of detected eyes and mouth. Figure 4.5(b) shows the detected feature points.



(a) (b)

(c) (d) (e)

Figure 4.4. Facial measurements of a human face: (a) color image; (b) range map; and the range map with texture mapped for (c) a left view; (d) a profile view; (e) a right view.

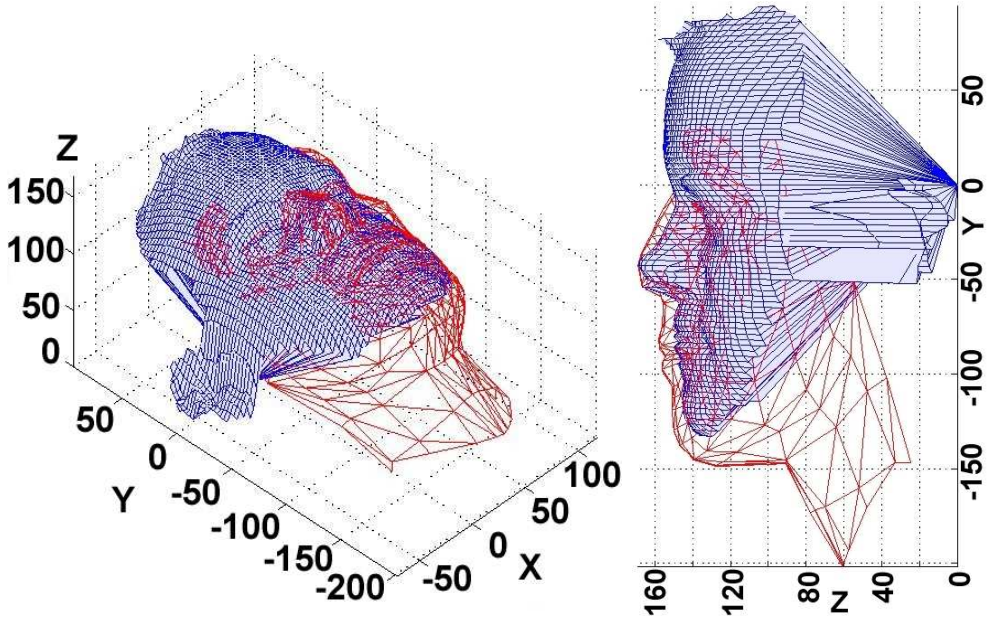


(a) (b)

Figure 4.5. Facial features overlaid on the color image, (a) obtained from face detection; (b) generated for face modeling.

4.4 Model Construction

Our face modeling process consists of *global alignment* and *local adaptation*. Global alignment first brings the generic model and facial measurements into the same coordinate system. Based on the 3D head pose and the face size, the generic model is then scaled, rotated, and translated to fit the facial measurements. Figure 4.6 shows the global alignment results in two different modes. Local adaptation consists of *local*



(a) (b)

Figure 4.6. Global alignment of the generic model (in red) to the facial measurements (in blue): the target mesh is plotted in (a) for a hidden line removal mode for a side view; (b) for a see-through mode for a profile view.

alignment and *local feature refinement*. Local alignment involves scaling and translating model features, such as eyes, nose, mouth, chin and face boundary to fit the extracted facial features. Local feature refinement makes use of two new techniques– *displacement propagation* and *2.5D active contours*–to smooth the face model and to refine local features. The local alignment and the local refinement of each feature

(shown in Fig. 4.2(c)) are followed by displacement (of model vertices) propagation, in order to blend features in the face model.

Displacement propagation inside a triangular mesh mimics the transmission of message packets in computer networks. Let N_i be the number of vertices that are connected to a vertex V_i, J_i be the set of all the indices of vertices that are connected to the vertex V_i, w_i be the sum of weights (each of which is the Euclidean distance between two vertices) on all the vertices that are connected to the vertex V_i, and d_{ij} be the Euclidean distance between the vertex V_i and a vertex V_j. Let ΔV_j be the displacement of vertex V_j, and α be the decay factor, which can be determined by the face size and the size of the active facial feature in each coordinate. Eq. (4.1) computes the contribution of vertex V_j to the displacement of vertex V_i.

$$\Delta V_{ij} = \begin{cases} \Delta V_j \cdot \frac{w_i - d_{ij}}{w_i \cdot (N_i - 1)} \cdot e^{-\alpha d_{ij}}, & N_i > 1, \quad w_i = \sum_{j \in J_i} d_{ij} \\ \Delta V_j \cdot e^{-\alpha d_{ij}}, & N_i = 1, \quad j \in J_i \ . \end{cases} \tag{4.1}$$

In other words, ΔV_{ij} is computed as the product of the displacement, the weight, and

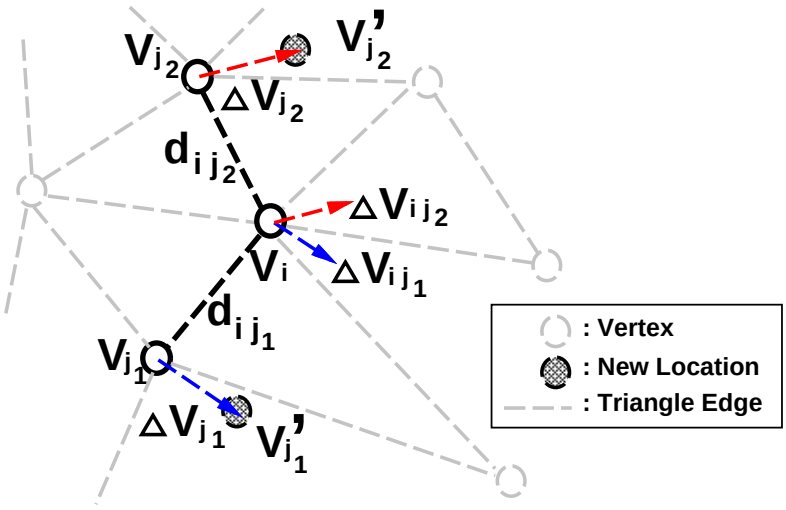
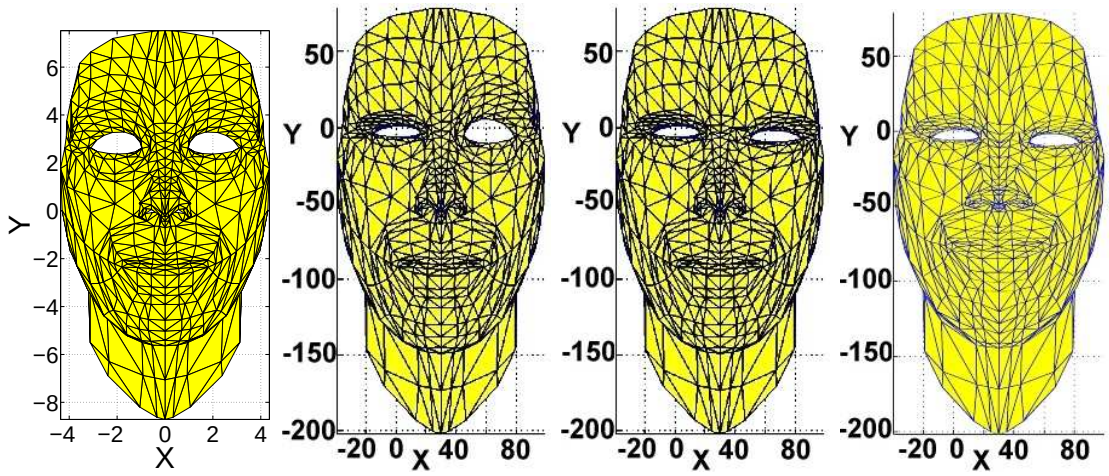


Figure 4.7. Displacement propagation.

a feature-dependent decay factor. Figure 4.7 depicts a small portion of a triangular mesh network around the vertex V_i. The mesh network illustrates the displacement, ΔV_{ij_1}, contributed by a vertex V_{j_1} (in blue), and the displacement, ΔV_{ij_2}, contributed by a vertex V_{j_2} (in red). In this case, the vertex V_i has six neighboring vertices, i.e., N_i is 6. The total displacement ΔV_i of V_i can be obtained by summing up all the displacements contributed by its neighboring vertices as follows.

$$\Delta V_i = \sum_{j \in J_i} \Delta V_{ij}.$$

The displacement will decay during propagation and it will continue for few iterations. The number of iterations is determined by the number of edge connections from the current feature to the nearest neighboring feature. In future implementations, we will include the symmetric property of a face and facial topology in computing this displacement. Figure 4.8 shows the results of local alignment for the frontal view after three iterations of displacement propagation.



(a) (b) (c) (d)

Figure 4.8. Local feature alignment and displacement propagation shown for the frontal view: (a) the input generic model; the model adapted to (b) the left eye; (c) the nose; (d) mouth and chin.

Local feature refinement follows local alignment to further adapt the aligned face model to an individual face by using 2.5D active contours (snakes). We modify Amini et al.'s [124] 2D snakes for our 3D active contours on boundaries of facial features. The active contours are useful for detecting irregular shapes by minimizing the (total) energy of the shape contour. The total energy, E_{total}, consists of the internal energy E_{int} (controlling the geometry of the contour) and external energy E_{ext} (controlling the desired shape). We reformulate the energy for our 3D snake as follows. Assume that an active contour includes a set of N vertices: $\{v_1, \cdots, v_{i-1}, v_i, v_{i+1}, \cdots, v_N\}$. The total energy can be computed by Eq. (4.2).

$$E_{\text{total}} = \sum_{i=1}^{N} \left[E_{\text{int}}(v_i) + E_{\text{ext}}(v_i) \right] \ . \tag{4.2}$$

The internal energy is listed in Eq. (4.3).

$$E_{\text{int}}(v_i) = \left(\alpha_i |v_i - v_{i-1}|^2 + \beta_i |v_{i+1} - 2v_i + v_{i-1}|^2 \right) /2, \tag{4.3}$$

where α_i controls the distance between vertices, and β_i controls the smoothness of the contours. The norm term $|\cdot|$ in Eq. (4.3) is determined by parameterized 3D coordinates, not merely 2D coordinates. Therefore, we call these contours *2.5D snakes*.

The initial contours needed for fitting the snakes are crucial. Fortunately, they can be obtained from our generic face model. Another important point for fitting snakes

is to find appropriate external energy maps that contain local maximum/minimum at the boundaries of facial features. For the face boundary and the nose, the external energy is computed by the maximum magnitude of vertical and horizontal gradients from range maps. These two facial features have steeper borders than others. For features such as eyes and the mouth, the external energy is obtained by a product of the magnitude of the luminance gradient and the squared luminance. Figure 4.9

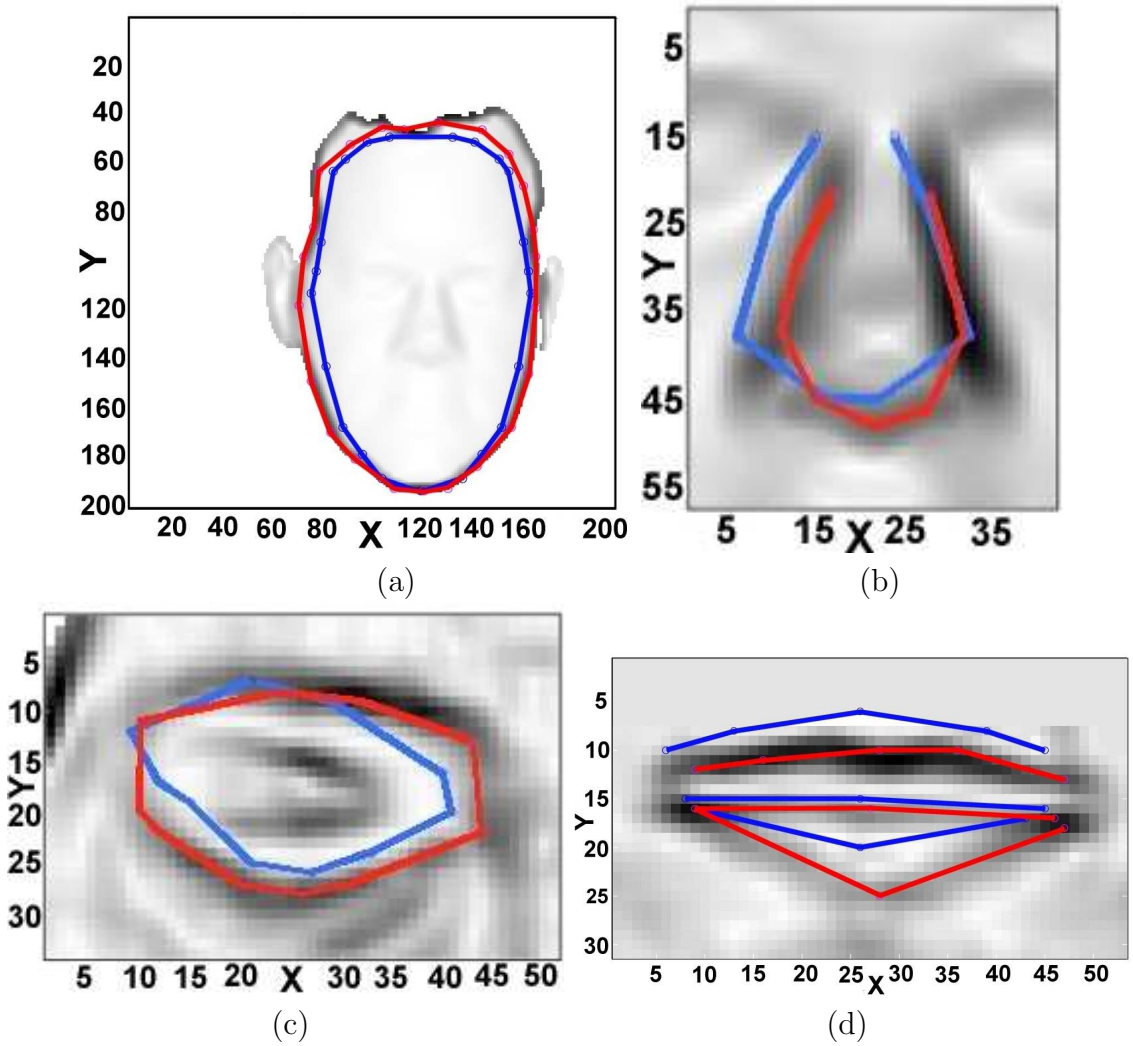
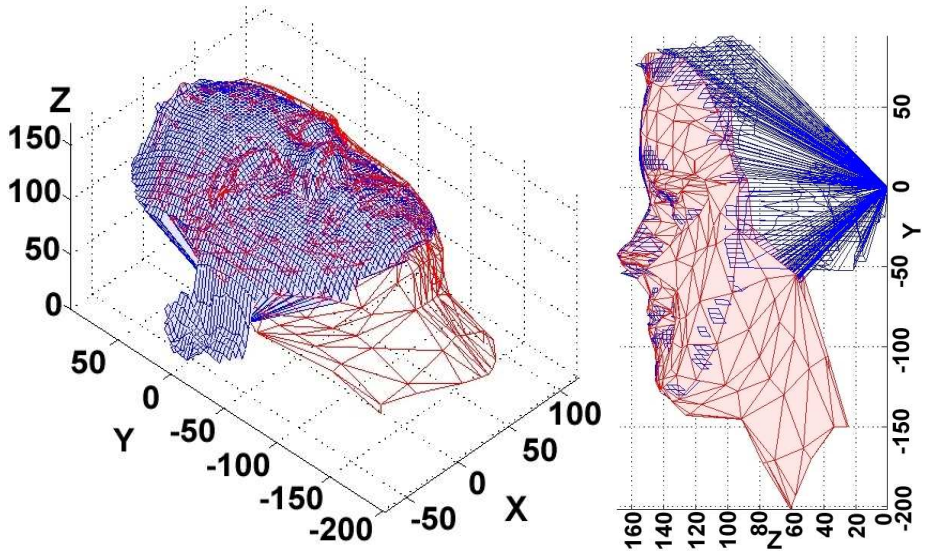


Figure 4.9. Local feature refinement: initial (in blue) and refined (in red) contours overlaid on the energy maps for (a) the face boundary; (b) the nose; (c) the left eye; and (d) the mouth.

shows the results of local refinement for the face boundary, the nose, the left eye, and the mouth.

Although our displacement propagation smoothes non-feature skin regions in the local adaptation, these skin regions can be further updated if a dense range map is available. However, based on our experiments, we find that the update of non-feature skin regions does not make a significant difference except in cheek regions because the displacement propagation already smoothes the skin regions surrounding each facial feature. Figure 4.10 shows the overlay of the final adapted face model in red and the target facial measurements in blue. For a comparison with Fig. 4.4, Fig. 4.11



(a) (b)

Figure 4.10. The adapted model (in red) overlapping the target measurements (in blue), plotted (a) in 3D; (b) with colored facets at a profile view.

shows the texture-mapped face model. The texture-mapped model is visually similar to the original face. We further use a face recognition algorithm [78] to demonstrate the use of 3D model. The training database contains (i) 504 images captured from 28 subjects and (ii) 15 images of one subject generated from our 3D face model,

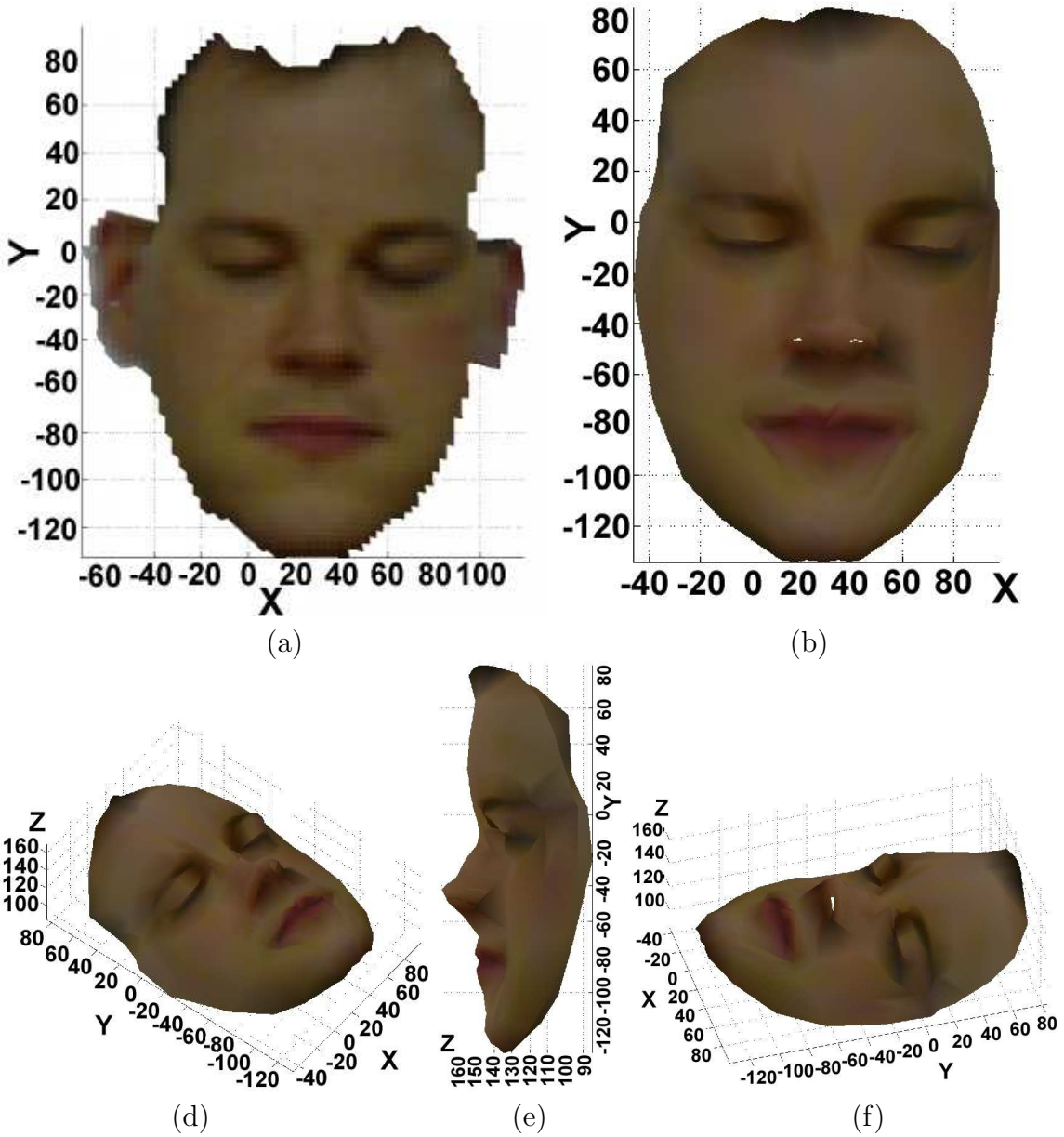


Figure 4.11. Texture Mapping. (a) The texture-mapped input range image. The texture-mapped adapted mesh model shown for (b) a frontal view; (d) a left view; (e) a profile view; (f) a right view.

which are shown in the top row in Fig. 4.12. All the 10 test images of the subject shown in the bottom row in Fig. 4.12 were correctly matched to our face model. This preliminary matching experiment shows that the proposed 3D face model is quite useful for recognizing faces at non-frontal views based on the facial appearance.

Figure 4.12. Face matching: the top row shows the 15 training images generated from the 3D model; the bottom row shows 10 test images of the subject captured from a CCD camera.

4.5 Summary

Face representation plays a crucial role in face recognition systems. For face recognition, we represent a human face as a 3D face model that is learned by adapting a generic 3D face model to input facial measurements in a global-to-local fashion. Based on the facial measurements, our model construction method first aligns the generic model globally, and then aligns and refines each facial feature locally using displacement (of model vertices) propagation and active contours associated with facial features. The final texture mapped model is visually similar to the original face. Initial matching experiments based on the 3D face model show encouraging results for appearance-based recognition.

Chapter 5

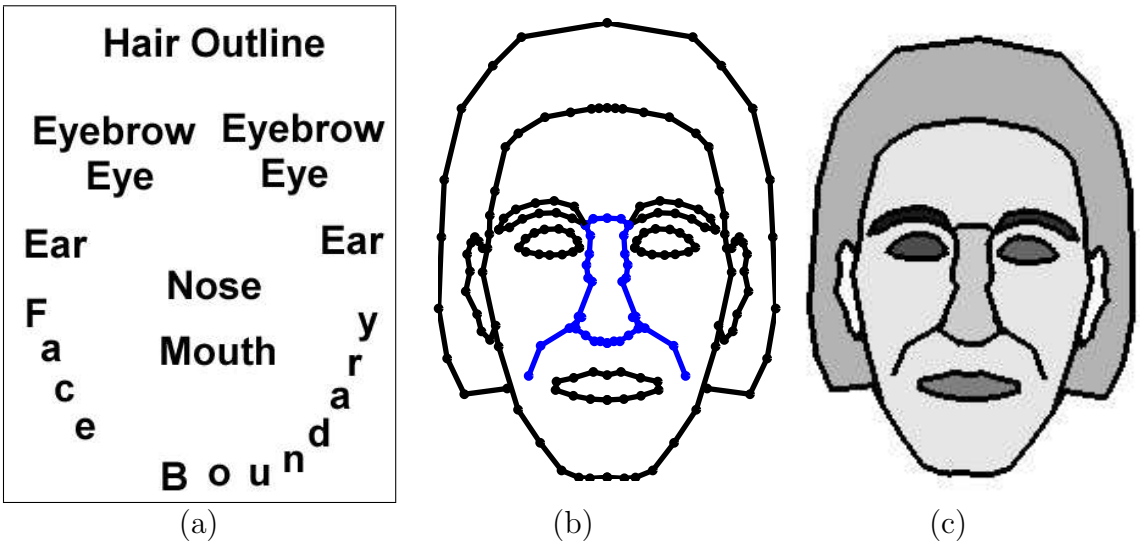
Semantic Face Recognition

In this chapter, we will describe semantic face matching (see Fig. 1.6 in Chapter 1) based on color input images and a generic 3D face model. We will give the details of (i) the face modeling from a single view (i.e., the frontal view), called face alignment, and (ii) recognition module in the semantic face matching algorithm. Section 5.1 describes the concept of semantic facial components, the semantic face graph, the generic 3D face model, and interacting snakes (multiple snakes that interact with each other). Section 5.2 describes the coarse alignment between the semantic graph and the input image based on the results of face detection. Section 5.3 presents the process of fine alignment of the semantic graph using interacting snakes. We explain how to compute the matching scores for graph alignment, and then show the resultant facial sketches and cartoon faces. Section 5.4 describes a semantic face matching method for recognizing faces, the use of component weights based on alignment scores, and the cost function for face identification. Then we give the algorithm of the proposed semantic face matching. We illustrate the generated cartoon faces from aligned semantic face

graphs. We demonstrate the experiment results on face matching based on a subset of the MPEG7 content set [15] and Michigan State University (MSU) face database. Section 5.5 describes the generation of facial caricatures, and discusses the effects of caricature on face recognition. A summary is given in Section 5.6.

5.1 Semantic Face Graph as Multiple Snakes

A semantic face graph provides a high-level description of the human face. A semantic graph projected onto a frontal view for face recognition is shown in Fig. 5.1. The nodes of the graph represent semantic facial components (e.g., eyes, mouth, and hair), each of which is constructed from a subset of vertices of the 3D generic face model and is enclosed by parametric curves. A semantic graph is represented in a 3D space and is compared with other such graphs in a 2D projection space. Therefore, the 2D appearance of the semantic graph looks different at different viewpoints due to the effect of perspective projection of the facial surface. We adopt Waters' animation model [69], [176] as the generic face model because it contains all the internal facial components, face outline, and muscle models for mimicking facial expressions. However, Waters' model does not include some of the external facial features, such as ears and hair. The hairstyle and the face outline play a crucial role in face recognition. Hence, we have created external facial components such as the ear and the hair contours for the frontal view of Waters' model. We hierarchically decompose the vertices of the mesh model into three levels: (i) vertices at the boundaries of facial components, (ii) vertices constructing facial components, and (iii) vertices belonging to facial skin

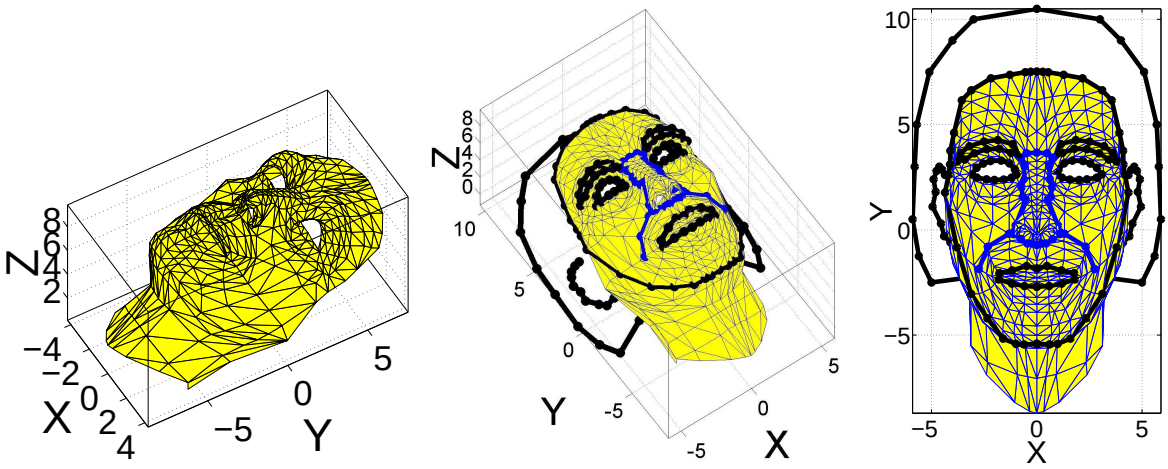


(a) (b) (c)

Figure 5.1. Semantic face graph is shown in a frontal view, whose nodes are (a) indicated by text; (b) depicted by polynomial curves; (c) filled with different shades. The edges of the semantic graph are implicitly stored in a 3D generic face model and are hidden here.

regions. The vertices at the top level are labelled with facial components such as the face outline, eyebrows, eyes, nose, and mouth (see Fig. 5.2). Let $\mathbf{T_0}$ denote the set of all semantic facial components, which are nodes of the generic semantic graph, $\mathbf{G_0}$. That is $T_0 = \{\{\text{left eyebrow}\}, \{\text{right eyebrows}\}, \{\text{left eye}\},..., \{\text{hair boundary}\}\}$. Let T be a subset of T_0, that is $T \subset 2^{T_0}$. Let M be the number of facial components in T. For example, T can be specified as $\{\{\text{left eye}\}, \{\text{right eye}\}\}, \{\text{mouth}\}\}$, where M is 3. Let the semantic graph projected on a 2D image, represented by the set $\mathbf{T}$, be $\mathbf{G}$. The coordinates of component boundary of $\mathbf{G}$ can be represented by a pair of sequences $x_i(n)$ and $y_i(n)$, where $n = 0, 1, \ldots, N_i - 1$ and $i = 1, \ldots, M$, for component i with N_i vertices. The 1D Fourier transform, $a_i(k)$, of the complex signal $u_i(n) = x_i(n) + jy_i(n)$ (where $j = \sqrt{-1}$) is computed by

$$a_i(k) = \mathcal{F}\{u_i(n)\} = \sum_{n=0}^{N_i-1} u_i(n) \cdot e^{-j2\pi kn/N_i}, \tag{5.1}$$



(a) (b) (c)

Figure 5.2. 3D generic face model: (a) Waters' triangular-mesh model shown in the side view; (b) model in (a) overlaid with facial curves including hair and ears at a side view; (c) model in (b) shown in the frontal view.

for facial component i with a close boundary such as eyes and mouth, and with end-vertex padding for those having open boundary such as ears and hair components. The advantage of using semantic graph descriptors for face matching is that these descriptors can seamlessly encode geometric relationships (scaling, rotation, translation, and shearing) among facial components in a compact format in the spatial frequency domain, because the vertices of all the facial components are specified in the same coordinate system with the origin around the nose (see Fig. 5.2). The reconstruction of semantic face graphs from semantic graph descriptors is obtained by

$$\tilde{u}_i(n) = \mathcal{F}^{-1}\{a_i(k)\} = \sum_{k=0}^{L_i-1} a_i(k) \cdot e^{j2\pi kn/N_i}, \tag{5.2}$$

where L_i ($< N_i$) is the number of frequency components used for the i^{th} face component. Figure 5.3 shows the reconstructed semantic face graphs at different levels

of Fourier series truncation. In addition, the coordinates of component boundary

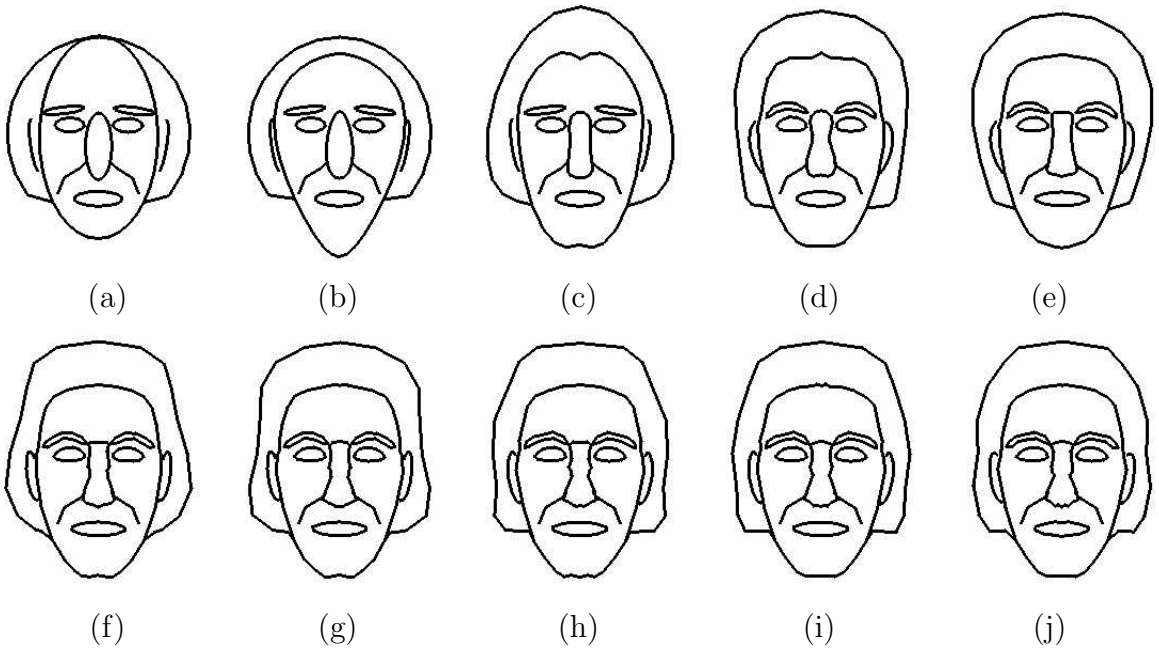


Figure 5.3. Semantic face graphs for the frontal view are reconstructed using Fourier descriptors with spatial frequency coefficients increasing from (a) 10% to (j) 100% at increments of 10%.

of $\mathbf{G}$ can also be represented by parametric curves, i.e., $c(s) = (x(s), y(s))$, where $s \in [0, 1]$, for explicit curve deformation or for generating level-set functions for implicit curve evolution. Therefore, the component boundaries of a semantic face graph are associated with a collection of active contours (snakes).

5.2 Coarse Alignment of Semantic Face Graph

Our face recognition system contains four major modules: face detection, pose estimation, face alignment, and face matching. The face detection module finds locations of face and facial features in a color image using the algorithm in [175]. Figures 5.4(a) to 5.4(d) show input color images and the results of face detection. Currently, we as-

sume that the face images have been captured at near frontal views (i.e., all of internal and external facial components are visible). The face alignment module makes use of the face detection results to align a semantic face graph onto the input image. The face alignment can be decomposed into the coarse and the fine alignment modules. In the coarse alignment, a semantic face graph at an estimated pose is aligned with

(a) (b) (c) (d)

Figure 5.4. Face detection results: (a) and (c) are input face images of size 640×480 from the MPEG7 content set; (b) and (d) are detected faces, each of which is described by an oval and a triangle.

a face image through the *global and local* geometric transformation (scaling, rotation, and translation), based on the detected locations of face and facial components. Section 5.3 will describe in detail the fine alignment, in which the semantic face graph is *locally* deformed to fit the face image.

Coarse alignment involves a rigid 3D transformation of the entire semantic graph. The parameters used in the transformation (scaling, rotation, and translation) are estimated from the outputs of the face detection algorithm. Besides the use of face detection results, we further employ the edges and color characteristics of facial components to locally refine the rotation, translation, and scaling parameters for individual components. This parameter refinement is achieved by maximizing a *semantic facial score* (SFS) through a small amount of perturbations of the parameters. The semantic face score takes into account the fitness of component boundary and of com-

ponent color. The semantic facial score of the set T on a face image $I(u,v)$, SFS_T, is defined by prior weights on facial components and component matching scores as follows:

$$SFS_T = \frac{\sum_{i=0}^{N-1} wt(i) \cdot MS(i)}{\sum_{i=0}^{N-1} wt(i)} - \rho \cdot SD\left(MS(i)\right), \tag{5.3}$$

where N is the number of semantic components, $wt(i)$ and $MS(i)$ are, respectively, the a priori weight and the matching score of component i, ρ is a constant used to penalize the components with high standard deviations of the matching scores, and $SD(x)$ stands for standard deviation of x.

The matching score for the i^{th} facial component is computed based on the coherence of the boundary and the coherence of color content (represented by a component map) by

$$\begin{aligned} MS(i) = & \frac{1}{M_i} \sum_{j=0}^{M_i-1} \left(\frac{1}{A_i} \sum_{k=0}^{A_i-1} e(u_k, v_k) \right) \cdot \\ & \frac{\left| cos(\theta_i^G(u_j, v_j) - \theta(u_j, v_j)) \right| + f(u_j, v_j)}{2}, \end{aligned} \tag{5.4}$$

where M_i and A_i are, respectively, the number of pixels along the curve of component i and those of pixels covered by the component i, θ_i^G and θ_i are the normal direction of component curve i in a semantic graph G and the gradient orientation of the image I, f is the edge magnitude of the image I, and $e(u_k, v_k)$ is the facial component map of the image I at pixel k. The gradients are computed as follows:

$$f(u_j, v_j) = \sum_{s=0}^{S} |\nabla G_{\sigma_s}(u_j, v_j) \circledast Y(u_j, v_j)| \quad (5.5)$$

$$\theta(u_j, v_j) = \sum_{s=0}^{S} \arg\left(\nabla G_{\sigma_s}(u_j, v_j) \circledast Y(u_j, v_j)\right) , \quad (5.6)$$

where Y is the luma of the color image I, and G_{σ_s} is the Gaussian function with zero mean and standard deviation σ_s. The largest standard deviation σ_S is limited by the distance between eyes and eyebrows where $S = 4$, and ∇ and $\circledast$ are the gradient and convolution operators. The gradient magnitude, gradient orientation, eye map [175] and coarse alignment results for the subject in Fig. 5.4(a) are shown in Fig. 5.5. The eye map is an average of a symmetry map [177] and an eye energy map (will be explained in Section 5.3.1). Furthermore, we construct a shadow map of a face image in order to locate eyebrow, nostril, and mouth lines, based on the average value of luminance intensity on a facial skin region (i.e., rectangles shown in Figs. 5.6(a) and 5.6(c)). These feature lines, shown as dark lines in Figs. 5.7(c), are used to adjust corresponding facial components of a semantic graph. Fig. 5.7 shows five examples of coarse alignment.

5.3 Fine Alignment of Semantic Face Graph via Interacting Snakes

Fine alignment employs active contours to locally refine facial components of a semantic face graph that is drawn from a 3D generic face model. The 2D projection of

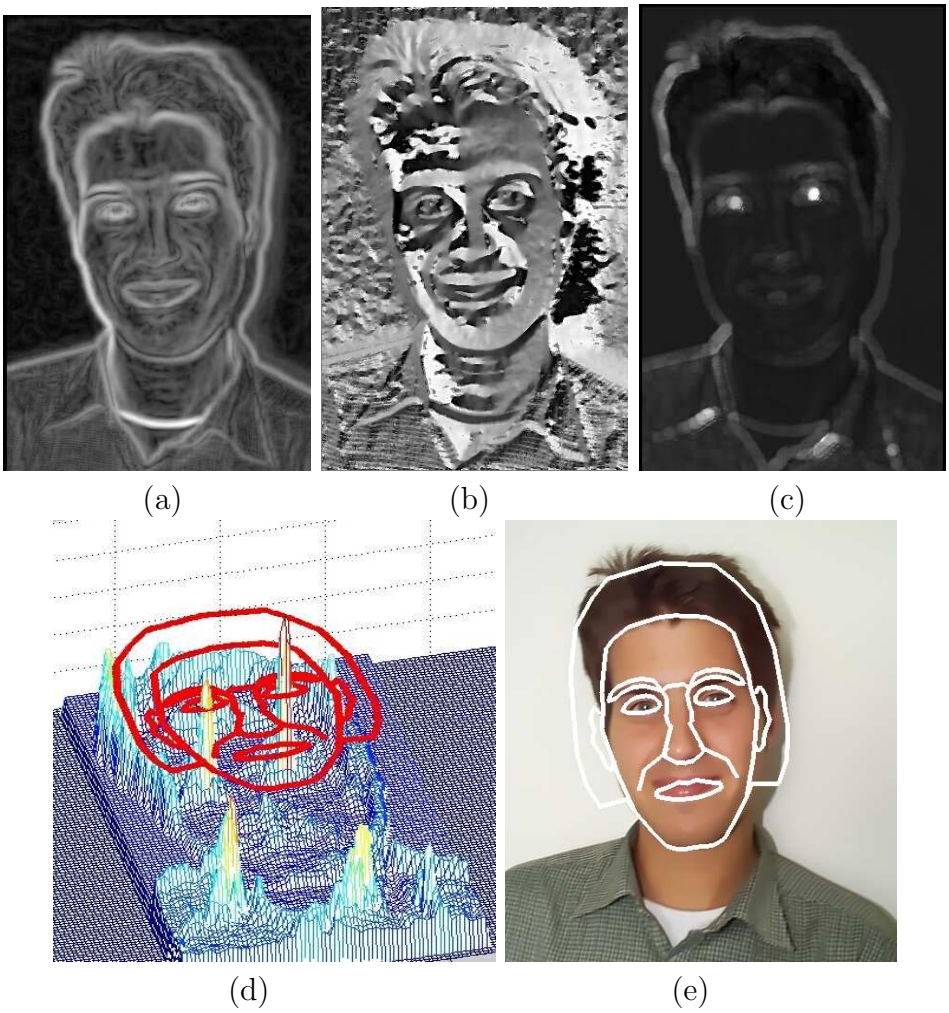


Figure 5.5. Boundary map and eye component map for coarse alignment: (a) and (b) are gradient magnitude and orientation, respectively, obtained from multi-scale Gaussian-blurred edge response; (c) an eye map extracted from a face image shown in Fig. 5.4(c); (d) a semantic face graph overlaid on a 3D plot of the eye map; (e) image overlaid with a coarsely aligned face graph.

a semantic face graph produces a collection of component boundaries, each of which is described by a closed (or open) active contour. The collection of these active contours, called *interacting snakes*, interact with each other through a repulsion energy in order to align the general facial topology onto the sensed face images in an iterative fashion. We have studied two competing implementations of active contours for the deformation of interacting snakes: (i) explicit (or parametric) and (ii) implicit contour

(a) (b) (c) (d)

Figure 5.6. Shadow maps: (a) and (c) are luma components of face images in Figs. 5.4(a) and 5.4(c), overlaid with rectangles within which the average values of skin intensity is calculated; (b) and (d) are shadow maps where bright pixels indicate the regions that are darker than average skin intensity.

representations. The explicit contour representation has the advantage of maintaining the geometric topology. The implicit contour representation requires topological constraints on the implicit function.

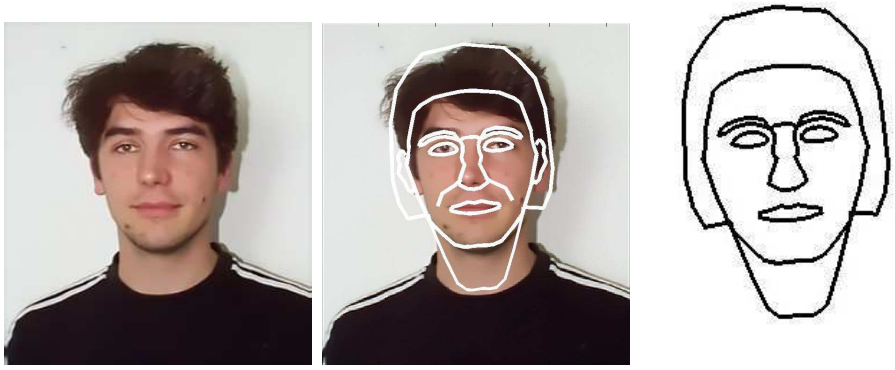
5.3.1 Interacting Snakes and Energy Functional

Active contours have been successfully used to impose high-level geometrical constraints on low-level features that are extracted from images. Active contours are iteratively deformed based on the initial configuration of the contours and the energy functional that is to be minimized. The initial configuration of interacting snakes is obtained from the coarsely-aligned semantic face graph, and is shown in Fig. 5.8(c). Currently, there are eight snakes interacting with each other. These snakes describe the hair outline, face outline, eyebrows, eyes, nose, and mouth of a face; they are denoted as $V(s) = \bigcup_{j=1}^{N} \{v_i(s)\}$, where N $(= 8)$ is the number of snakes, and $v_i(s)$ is the i^{th} snake with the parameter $s \in [0, 1]$.

The energies used for minimization include the internal energy of a contour (i.e., smoothness and stiffness energies), and the external energy (i.e., the inverse of edge

(a) (b) (c) (d)

Figure 5.7. Coarse alignment: (a) input face images of size 640×480 from the MPEG7 content set (first three rows), and of size 256×384 from the MSU database (the fourth row); (b) detected faces; (c) locations of eyebrow, nostril, and mouth lines using shadow maps; (d) face images overlaid with coarsely aligned face graphs.



(a) (b) (c)

Figure 5.8. Interacting snakes: (a) face region extracted from a face image shown in Fig. 5.4(a); (b) image in (a) overlaid with a (projected) semantic face graph; (c) the initial configuration of interacting snakes obtained from the semantic face graph shown in (b).

strength) extracted from an image. In addition to minimizing the internal energy of an individual curve, interacting snakes minimize the attraction energy on both the contours and enclosed regions of individual snakes, and the repulsion energy among multiple snakes. The energy functional used by interacting snakes is described in Eq. (5.7).

$$E_{isnake} = \sum_{i=1}^{N} \left[\int_0^1 \underbrace{E_{internal}(v_i(s)) + E_{repulsion}(v_i(s))}_{E_{prior}} + \underbrace{E_{attraction}(v_i(s))}_{E_{observation}} ds \right], \tag{5.7}$$

where i is the index of the interacting snake. The first two energy terms are based on the prior knowledge of snake's shape and snakes' configuration (i.e., facial topology) while the third energy term is based on the sensed image (i.e., observed pixel values).

In the Baysian framework, given an image I, minimizing the energy of interacting snakes is equivalent to maximizing a posteriori probability $p(V|I)$ of interacting snakes $V(s)$ with a 0/1 loss function:

$$p(V|I) = \frac{p(I|V) \cdot p(V)}{p(I)}, \tag{5.8}$$

where $p(I|V) \sim e^{-E_{obersvation}}$, $p(V) \sim e^{-E_{prior}}$, $p(V)$ is the prior probability of snakes' structure and $p(I|V)$ is the conditional probability of the image potential of interacting snakes. From calculus of variations, we know that interacting snakes which minimize the energy function in Eq. (5.7) must satisfy the following Euler-Lagrange equation:

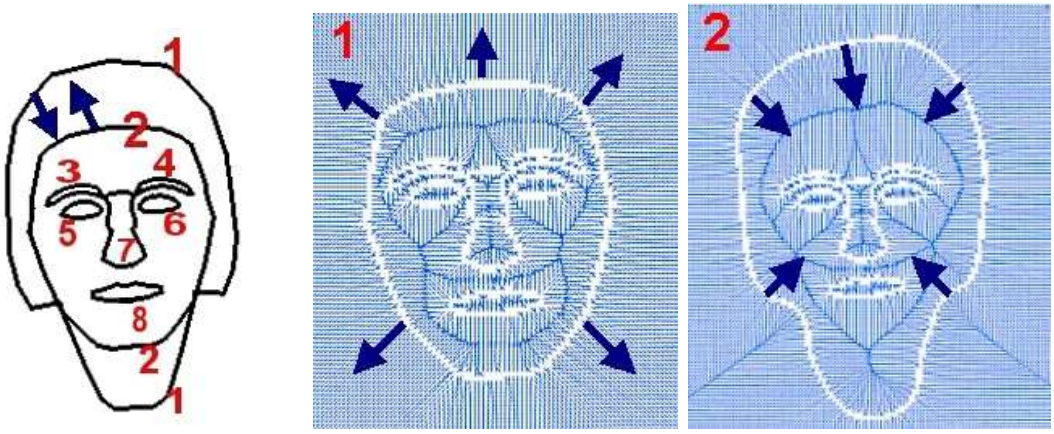
$$\sum_{i=1}^{N} \left[\underbrace{\alpha v_i''(s) - \beta v_i^{(4)}(s)}_{\text{Internal Force}} \underbrace{-\nabla E_{repulsion}(v_i(s))}_{\text{Repulsion Force}} \underbrace{-\nabla E_{attraction}(v_i(s))}_{\text{Attraction Force}} \right] = 0, \tag{5.9}$$

where α and β are coefficients for adjusting the second- and the fourth- order derivatives of a contour, respectively. Repulsion force field is constructed based on the gradients of distance map among the interacting snakes as follows:

$$-\nabla E_{repulsion}(v_i(s)) = \nabla \left(\left[EDT(\bigcup_{j=1, j\neq i}^{N} v_j(s)) \right]^2 \right), \tag{5.10}$$

where EDT is a signed Euclidean Distance Transform [178]. Figure 5.9 show the repulsion force fields for the hair outline and the face outline. The use of the repulsion force can prevent different active contours from converging to the same locations of

minimum energy. The attraction force field consists of two kinds of fields in Eq.



(a) (b) (c)

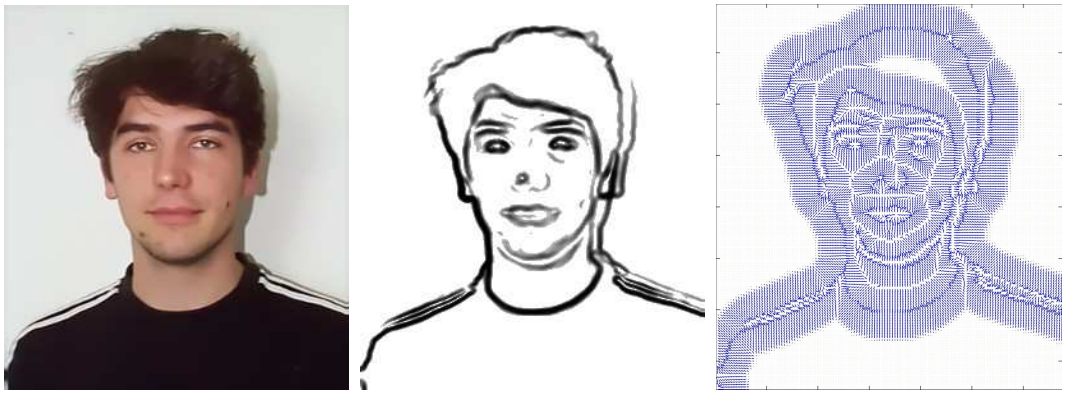
Figure 5.9. Repulsion force: (a) interacting snakes with index numbers marked; (b) the repulsion force computed for the hair outline; (c) the repulsion force computed for the face outline.

(5.11): one is obtained from edge strength, called gradient vector field (GVF) [127], and the other from a region pressure field (RPF) [133].

$$\begin{aligned} -\nabla E_{image}(v_i(s)) &= GVF + RPF \\ &= GVF + \rho \cdot \vec{N}(v_i(s)) \cdot \left(1 - \frac{|E_i^{comp}(v_i(s)) - \mu|}{k\sigma}\right), \end{aligned} \tag{5.11}$$

where $\vec{N}(v_i(s))$ is the normal vector on the i^{th} contour $v_i(s)$; E_i^{comp} is the component energy of the i^{th} component; μ, σ are the mean and the standard deviation of region energy over a seed region of the i^{th} component; k is a constant that constrains the energy variation of a component. The advantage of using GVF for snake deformation is that its range of influence is larger than that obtained from gradients, and can attract snakes to a concave shape. A GVF is constructed from an edge map by an iterative process. However, the construction of GVF is very sensitive to noise in the

edge map; hence it requires a clean edge map as an input. Therefore, we compute a GVF by using three edge maps obtained from luma and chroma components of a color image, and by choosing as the edge pixels the top $p\%$ (= 15%) of edge pixel population over a face region, as shown in Fig. 5.10(a). Figure 5.10(b) is the edge map for constructing its GVF that is shown in Fig. 5.10(c). The region



(a) (b) (c)

Figure 5.10. Gradient vector field: (a) face region of interest extracted from a 640x480 image; (b) thresholded gradient map based on the population of edge pixels shown as dark pixels; (c) gradient vector field.

pressure field is available only for a homogeneous region in the image. However, we can construct component energy maps that reveal the color property of facial components such as eyes with bright-and-dark pixels and mouth with red lips. Then a region pressure field can be calculated based on the component energy map and on the mean and standard deviation of the energy over seed regions (note that we know the approximate locations of eyes and mouth). Let a color image have color components in the RGB space denoted as (R, G, B), and those in YCbCr space as (Y, Cb, Cr). An eye component energy for a color image is computed as follows:

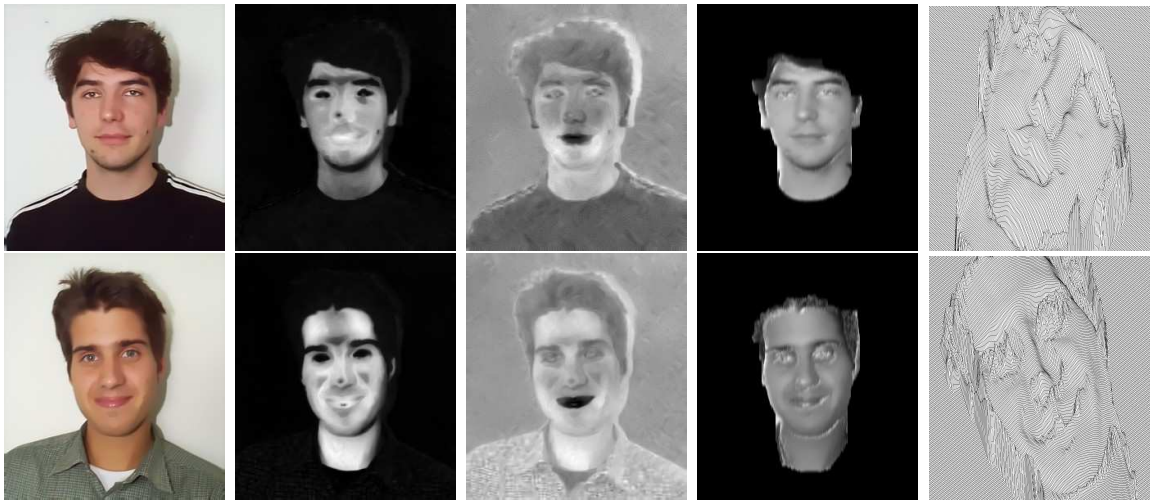
$$E_{eye}^{comp} = E_{msat} + E_{csh} + E_{cdif}, \tag{5.12}$$

$$E_{msat} = [\Bigg((R - \frac{K}{3})^2 + (G - \frac{K}{3})^2 + (B - \frac{K}{3})^2 - \frac{(R+G+B-K)^2}{3}\Bigg)^{0.5}], \tag{5.13}$$

$$E_{csh} = [[Cr - K/2]^2 - [Cb - K/2]^2], \tag{5.14}$$

$$E_{cdif} = [[Cr] - [Cb]], \tag{5.15}$$

where E_{msat} is the modified saturation (that is the distance in the plane between a point (R, G, B) and $(K/3, K/3, K/3)$) where $R + G + B = K$, E_{csh} is chroma shift, E_{cdif} is chroma difference, $K = 256$ is the number of grayscales for each color component, and $[x]$ indicates a function that normalizes x into the interval $[0, 1]$. The eye component energies for subjects in Fig. 5.11(a) is shown in Fig. 5.11(b). The



(a) (b) (c) (d) (e)

Figure 5.11. Component energy (darker pixels have stronger energy): (a) face region of interest; (b) eye component energy; (c) mouth component energy; (d) nose boundary energy; (e) nose boundary energy shown as a 3D mesh surface.

mouth component energy is computed as $E_{mouth}^{comp} = [-[Cb] - [Cr]]$. Figure 5.11(c)

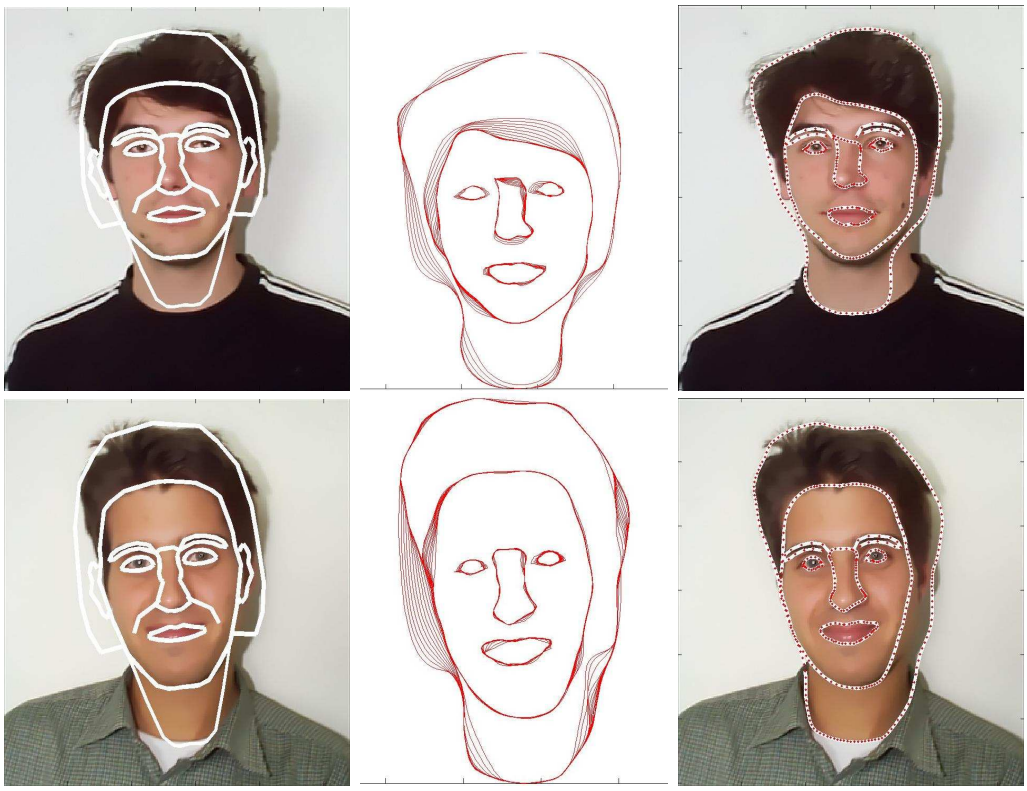
shows examples of mouth energies. For the nose component, its GVF is usually weak, and it is difficult to construct an energy map for nose. Hence, for the nose, we utilize Tsai and Shah's shape-from-shading (SFS) algorithm [179] to generate a boundary energy for augmenting the GVF for the nose component. The illumination direction used in the SFS algorithm is estimated from the average gradient fields of a face image [180] within a facial region. Figures 5.11(d) and 5.11(e) show examples of nose boundary energies in a 2D grayscale image and a 3D mesh plot, respectively.

5.3.2 Parametric Active Contours

Once we obtain the attraction force, we can make use of the implicit finite differential method [77], [127] and the iteratively updated repulsion force to deform the snakes. The stopping criteria is based on the iterative movement of each snake. Figure 5.12(a) shows the initial interacting snakes, Fig. 5.12(b) shows snake deformation without the eyebrow snakes, and Fig. 5.12(c) shows finely aligned snakes. Component matching scores in Eq. (5.4) are then updated based on the line and region integrals of boundary and component energies, respectively. We discuss another approach for deforming the interacting snakes based on geodesic active contours and level-set functions in Section 5.3.3.

5.3.3 Geodesic Active Contours

As implicit contours, geodesic snakes [181], which employ level-sets functions [182] are designed for extracting complex geometry. We initialize a level-set function using



(a) (b) (c)

Figure 5.12. Fine alignment: (a) snake deformation shown every five iterations; (b) aligned snakes (currently six snakes—hairstyle, face-border, eyes, and mouth—are interacting); (c) gradient vector field overlaid with the aligned snakes.

signed Euclidean distances from interacting snakes with positive values inside facial components such as hair, eyebrows, eyes, nose, mouth, and an additional neck component, Ω_i^+, where i is an integer, $i \in [1, 8]$; and with negative values over the facial skin and background regions, Ω_j^-, where j is either 1 or 2 . Different shades are filled in component regions, Ω_i^+ and Ω_j^-, to form a *cartoon face*, as shown in Fig. 5.13(c). Because facial components have different region characteristics, we modified Chan et al.'s approach [130] to take multiple regions and edges into account. The evolution step for the level-sets function,Φ, is described as follows:

$$\frac{\partial \Phi}{\partial t} = \nabla\Phi \left[\mu_1 \left(div(g\frac{\nabla\Phi}{|\nabla\Phi|}) - \mu_2 r - \alpha g \right) - \sum_i |I - c_i|^2 + \sum_j |I - c_j|^2 \right], \tag{5.16}$$

$$g = (1 - \frac{g_0}{\max(g_0)})^2, \quad g_0 = \log(1 + |\nabla I|^2)^2 \tag{5.17}$$

$$r = \begin{cases} 1/dt & dt \neq 0 \\ \text{MAXDIST} & dt = 0 \end{cases} \tag{5.18}$$

where μ_1 is a constant, μ_2 and α are constants in the interval between 0 and 1, I is the image color component, c_i and c_j are the average color components of facial component i over region Ω_i^+ and component j over Ω_j^-, respectively, r is the component repulsion, dt is the absolute Euclidean distance map of the face graph, and MAXDIST is the maximum distance in the image. We further preserve facial topology using topological numbers and the narrow band implementation of level-set functions [183]. The preliminary results are shown in Fig. 5.13 with evolution details and in Fig. 5.14 without the evolution details.

The facial distinctiveness of individuals can be seen from the changes among the generic, the fine fitted, and fine deformed face templates, shown in Figs. 5.13(a), 5.13(d), and 5.13(e). Comparing the two approaches for deforming interacting snakes, we believe that the first approach, parametric active contours, is better suited to the deformation of semantic face graphs.

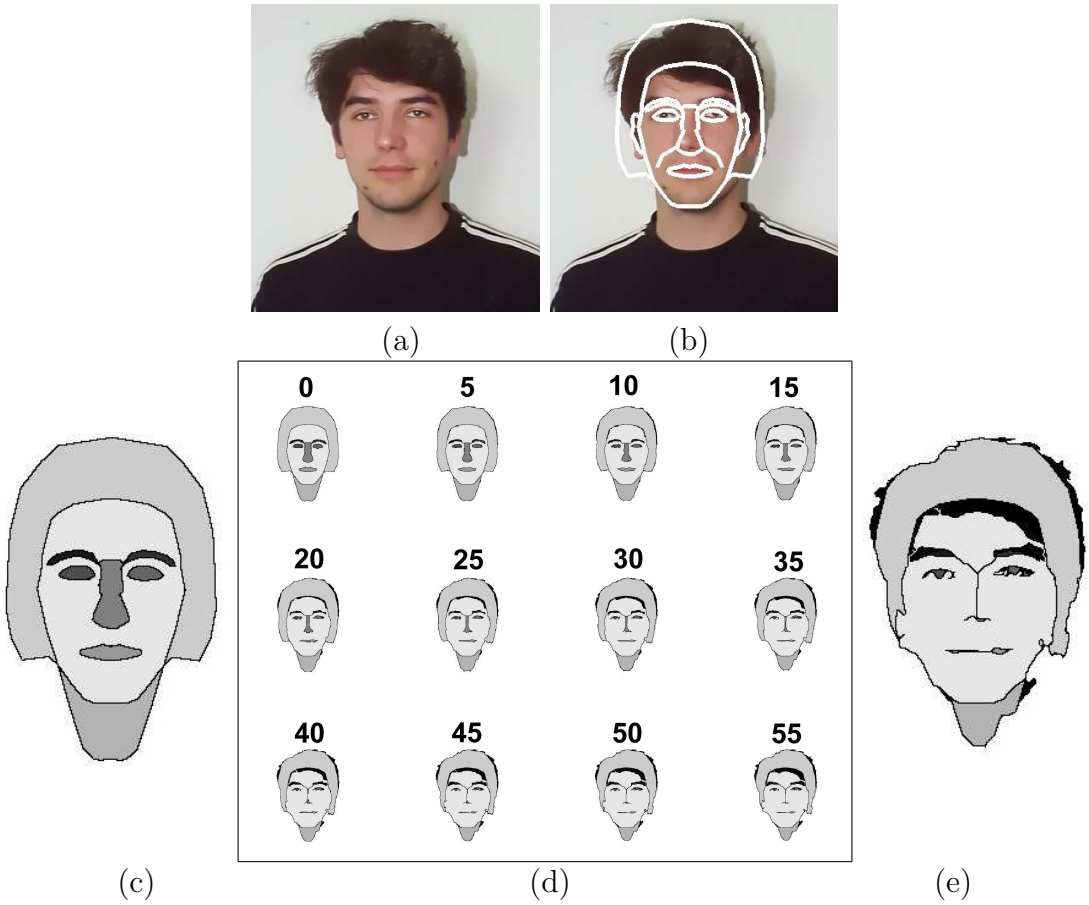


Figure 5.13. Fine alignment with evolution steps: (a) a face image; (b) the face in (a) overlaid with a coarsely aligned face graph; (c) initial interacting snakes with different shades in facial components (cartoon face); (d) curve evolution shown every five iterations (totally 55 iterations); (e) an aligned cartoon face.

5.4 Semantic Face Matching

We have developed a method to automatically derive semantic component weights for facial components based on coarsely aligned and finely deformed face graphs. These component weights are used to emphasize salient facial features for recognition (i.e., for computing a matching cost for a face comparison using semantic face graphs. The aligned face graph can also be used for generating facial caricatures.

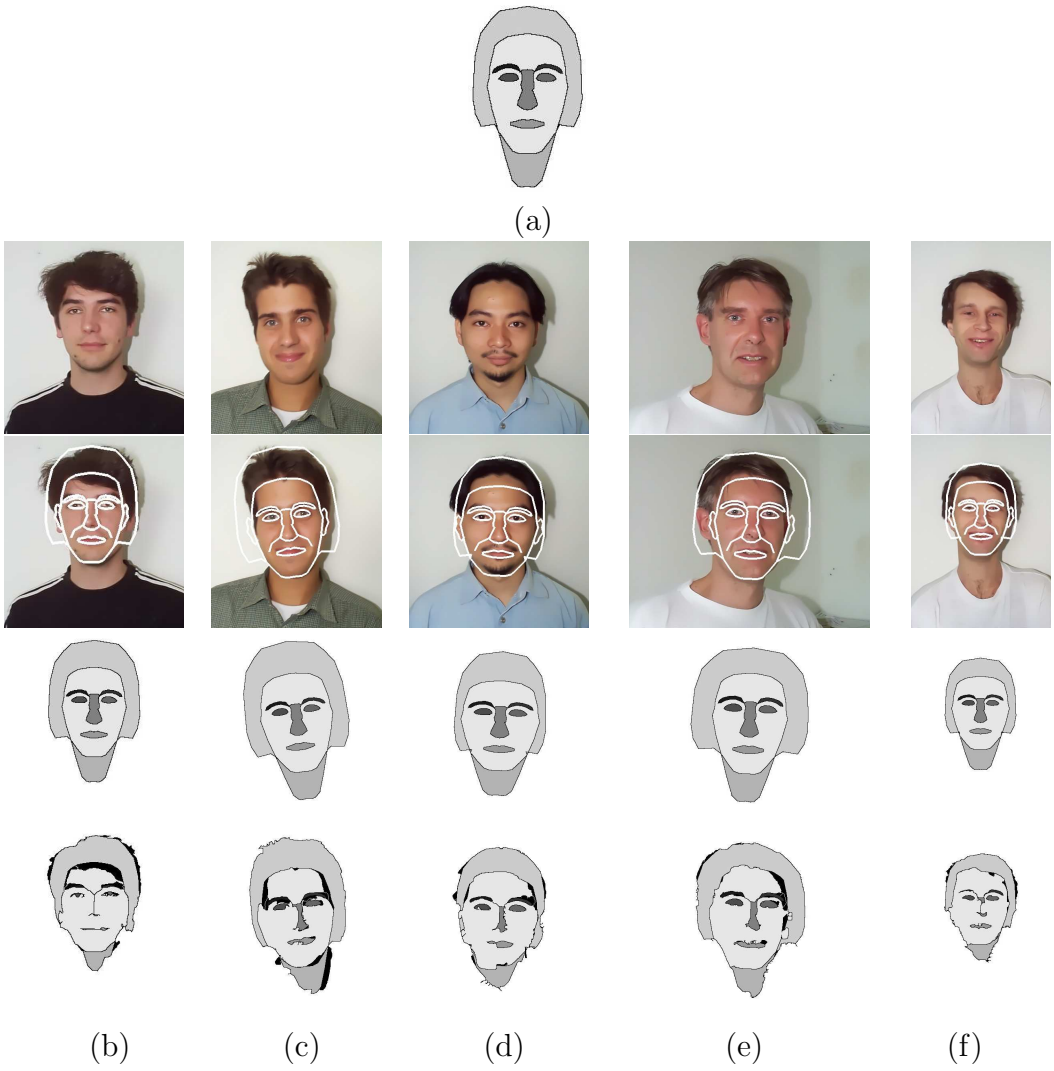


Figure 5.14. Fine alignment using geodesic active contours: (a) a generic cartoon face constructed from interacting snakes; (b) to (f) for five different subjects. For each subject, the image in the first row is the captured face image; the second row shows semantic face graphs obtained after coarse alignment, and overlaid on the color image; the third row shows semantic face graphs with individual components shown in different shades of gray; the last row shows face graphs with individual components after fine alignment.

5.4.1 Component Weights and Matching Cost

After the two phases of face alignment, we can automatically derive a weight (called *semantic component weight*) for each facial component i for a subject P with N_p training face images by

$$scw^P(i) = \begin{cases} 1 + e^{-2\sigma_d^2(i)/d^2(i)} & N_p > 1, \\ 1 + e^{-1/d^2(i)} & N_p = 1, \end{cases} \tag{5.19}$$

$$d(i) = \frac{1}{N_P} \sum_{k=1}^{N_P} SFD_i(\mathbf{G_0}, \mathbf{G_{P_k}}) \cdot MS^{P_k}(i), \tag{5.20}$$

$$\sigma_d(i) = SD_k \left[SFD_i(\mathbf{G_0}, \mathbf{G_{P_k}}) \cdot MS^{P_k}(i) \right], \tag{5.21}$$

where SFD means semantic facial distance, MS is the matching score, SD stands for standard deviation, $\mathbf{G_0}$ and $\mathbf{G_{P_k}}$ are the coarsely aligned and finely deformed semantic face graphs, respectively. The semantic component weights take values between 1 and 2. The semantic facial distance of facial component i between two graphs is defined as follows

$$\begin{aligned} SFD_i(\mathbf{G_0}, \mathbf{G_{P_k}}) &= Dist(SGD_i^{\mathbf{G_0}}, SGD_i^{\mathbf{G_{P_k}}}) \\ &= \left[\frac{1}{L_i} \sum_{k=0}^{L_i} \left| a_i^{\mathbf{G_0}}(k) - a_i^{\mathbf{G_{P_k}}}(k) \right|^2 \right]^{0.5}, \end{aligned} \tag{5.22}$$

where SGD stands for semantic graph descriptors. The distinctiveness of a facial component is evaluated by the semantic facial distance SFD between the generic semantic face graph and the aligned/matched semantic graph. The visibility of a facial component (due to head pose, illumination, and facial shadow) is estimated

by the reliability of component matching/alignment (i.e., matching scores for facial components). Finally, the 2D semantic face graph of subject P can be learned from N_p images under similar pose by

$$\mathbf{G_P} = \bigcup_i \mathcal{F}^{-1}\left\{\frac{1}{N_P}\sum_{k=1}^{N_P} SGD_i^{\mathbf{G_{P_k}}}\right\}. \tag{5.23}$$

The matching cost between the subject P and the k-th face image of subject Q can be calculated as

$$C(P, Q_k) = \sum_{i=1}^{M}\left\{scw^P(i)\cdot scw^{Q_k}(i)\cdot SFD_i(\mathbf{G_P}, \mathbf{G_{Q_k}})\right\}, \tag{5.24}$$

where M is the number of facial components. Face matching is accomplished by minimizing the matching cost.

5.4.2 Face Matching Algorithm

The system diagram of our proposed semantic face recognition method was illustrated in Fig. 1.6 (in Chapter 1). Figure 5.15 describes the semantic face matching algorithm for identifying faces with no rejection. The inputs of the algorithm are training images of M known subjects in the enrollment phase and one query face image of an unknown subject in the recognition phase. The query input can be easily generalized to either multiple images of an unknown subject or multiple images of multiple unknown subjects. Each known subject, P^j, can have $N^j(\geq 1)$ training images. The output of the algorithm is the identity of the unknown query face image(s) among M known subjects (a rejection option can be included by providing a threshold on the

matching cost in the algorithm). The algorithm uses our face detection method to locate faces and facial features in all the images, and the coarse and fine alignment methods to extract semantic facial features for face matching. Finally, it computes a matching cost for each comparison based on selected facial components, the derived component weights (distinctiveness), and matching score (visibility).

Figure 5.15. A semantic face matching algorithm.

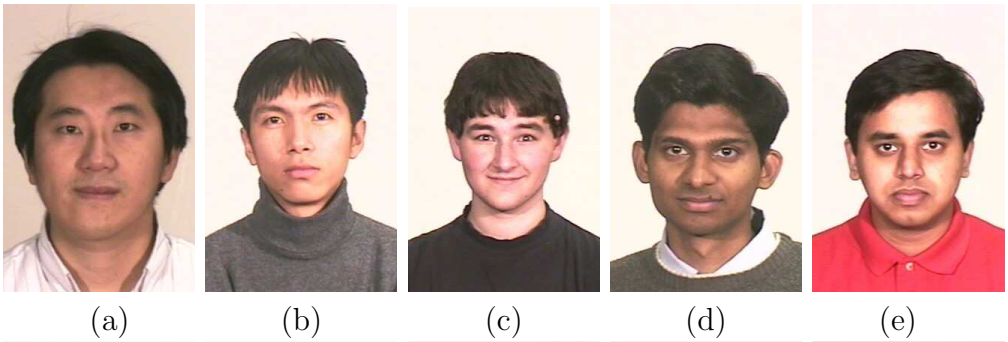
INPUT:	- N^j training face images for the subject P^j, $j = 1, \ldots, M$ - one query face image for unknown subject Q
Step 1:	Detect faces for all the images using the method in [175] $\longrightarrow$ Generate locations of face and facial features
Step 2:	Form a set of facial components, T, for recognition by assigning prior component weights
Step 3:	Coarsely align a generic semantic face graph to each image based on T $\longrightarrow$ Obtain component matching scores for each graph in Eq. (5.4)
Step 4:	Deform a coarsely-aligned face graph $\longrightarrow$ Update component matching scores based on integrals of component energies
Step 5:	Compute semantic facial descriptors SGD for each graph using the 1-D Fourier transform in Eq. (5.1).
Step 6:	Compute semantic component weights for each graph in Eqs. (5.19)-(5.21)
Step 7:	Integrate all the face graphs of subject P^j in Eq. (5.23), resulting in M template face graphs
Step 8:	Compute M matching costs, $C(P^j, Q_k)$, between P^j and Q_k in Eq. (5.24), where $k = 1$, $j = 1, \ldots, M$
Step 9:	Subject P^J with the minimum matching cost has the best matched face to the unknown subject Q_k.
OUTPUT:	$Q = P^J$

5.4.3 Face Matching

We have constructed a small face database of ten subjects (ten images per subject) at near frontal views with small amounts of variations in facial expression, face orien-

tation, face size, and lighting conditions, during different image capture sessions over a period of two months. Figure 5.16 shows five images of one subject, while Fig. 5.17 shows one image each of ten subjects.

Figure 5.16. Five color images (256×384) of a subject.



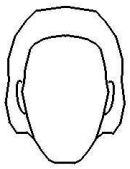
(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 5.17. Face images of ten subjects.

We employ 5 images each for 10 subjects for training the semantic face graphs. With re-substitution and leave-one-out tests, the misclassification rates are shown in Table 5.1 using different sets of facial components and semantic graph descriptors with the number of frequency components truncated at three levels. External facial

Table 5.1
Error rates on a 50-image database.

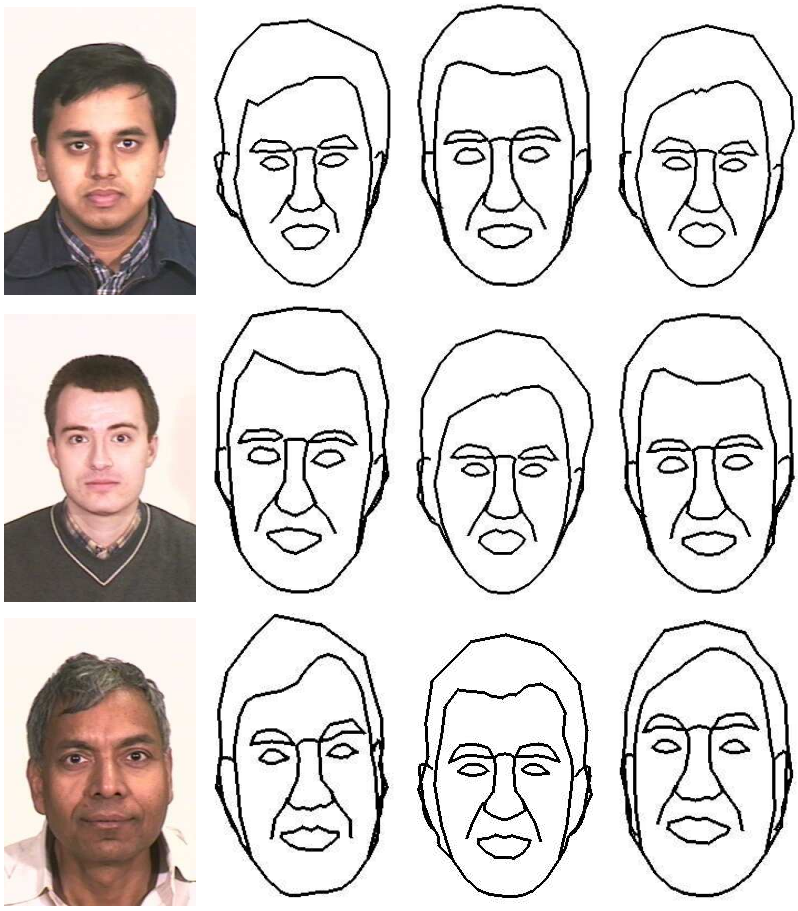
Component Set	T_1		T_2		T_3		T_4	
Face Graph								
P (%)	RS	LOO	RS	LOO	RS	LOO	RS	LOO
100%	0%	6%	0%	6%	12%	24%	16%	30%
50%	0%	6%	0%	6%	12%	24%	16%	30%
30%	0%	6%	0%	12%	16%	24%	18%	34%

P: % of frequency components, T_1: All components, T_2: External components, T_3: Internal components, T_4: Eyes and Eyebrows, RS: Re-substitution, LOO: Leave-one-out.

Table 5.2
Dimensions of the semantic graph descriptors for individual facial components.

P (%)	100%	50%	30%
Dimension	N_i	L_i	L_i
Eyebrow	12	5	3
Eye	13	7	3
Nose	34	13	7
mouth	14	7	3
Face outline	36	17	11
Ear	11	5	3
Hair	19	9	5

P: % of frequency components, N_i: the dimension of semantic graph descriptors, L_i: the dimension of truncated descriptors.



(a) (b) (c) (d)

Figure 5.18. Examples of misclassification: (a) input test image; (b) semantic face graph of the image in (a); (c) face graph of the misclassified subject; (d) face graph of the genuine subject obtained from the other images of the subject in the database (i.e., without the input test image in (a)). Each row shows one example of misclassification.

components include face outline, ears, and hairstyle, while internal components are eyebrows, eyes, nose, and mouth. We can see that the external facial components play an important role in recognition, and the Fourier descriptors provide compact features for classification because the dimensionality of our feature space is lower (see Table 5.2) compared to those used in eigen-subspace methods. Figure 5.18 shows the three examples of misclassification in a leave-one-out test for the facial component set T_1 using all the frequency components. The false matching may result from the

similar configuration of facial components, the biased average facial topology of the generic face model, and coarse head pose. Figures 5.19, 5.20, and 5.21 show the reconstructed semantic face graphs, $\mathbf{G_P}$ in Eq. (5.23), (compare them with $\mathbf{G_0}$ in Fig. 5.1(c)) at three levels of details, respectively. Each coarse alignment and fine

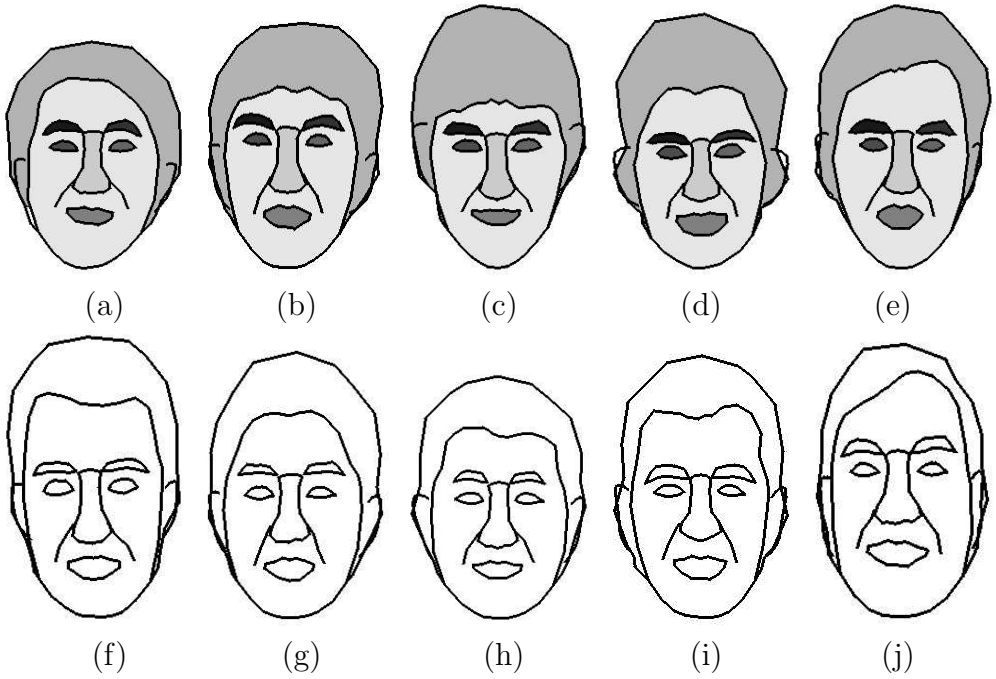


Figure 5.19. Cartoon faces reconstructed from Fourier descriptors using all the frequency components: (a) to (j) are ten average cartoon faces for ten different subjects based on five images for each subject. Individual components are shown in different shades in (a) to (e).

alignment on an image of size 640×480 takes 10 sec with C implementation and 460 sec. with MATLAB implementation, repsectively, while each face comparison takes 0.0029 sec with Matlab implementation on a 1.7 GHz CPU. We are conducting other cross-validation tests for classification, and are in the process of performing recognition on gallery (containing known subjects) and probe (containing unknown subject) databases. Although the alignment is off-line currently, there is large room to enhance the performance of alignment implementation to make it operate in real-

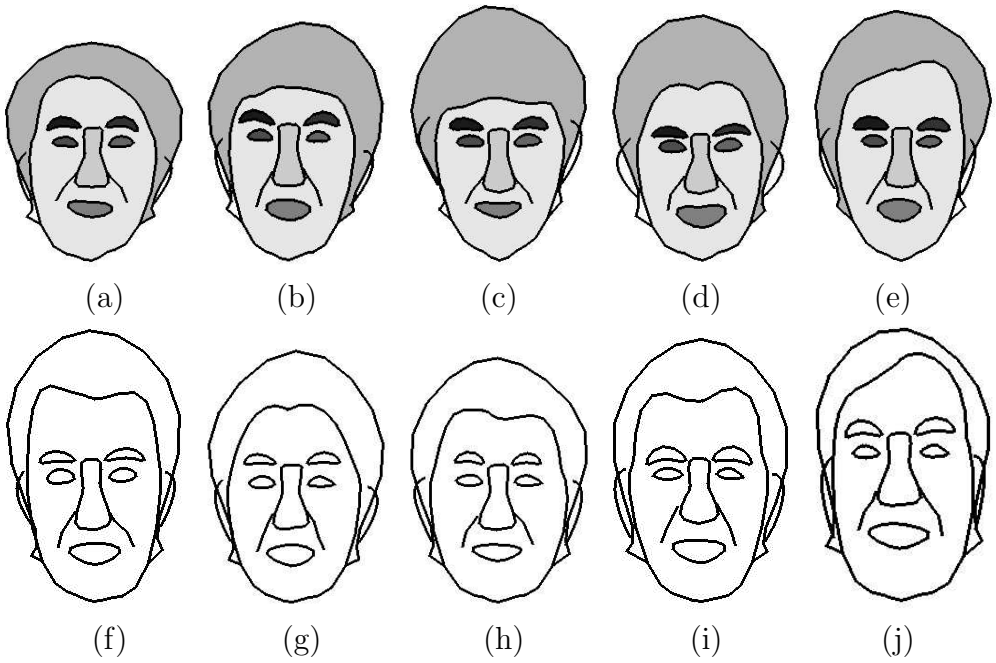


Figure 5.20. Cartoon faces reconstructed from Fourier descriptors using only 50% of the frequency components: (a) to (j) are ten average cartoon faces for ten different subjects based on five images for each subject. Individual components are shown in different shades in (a) to (e).

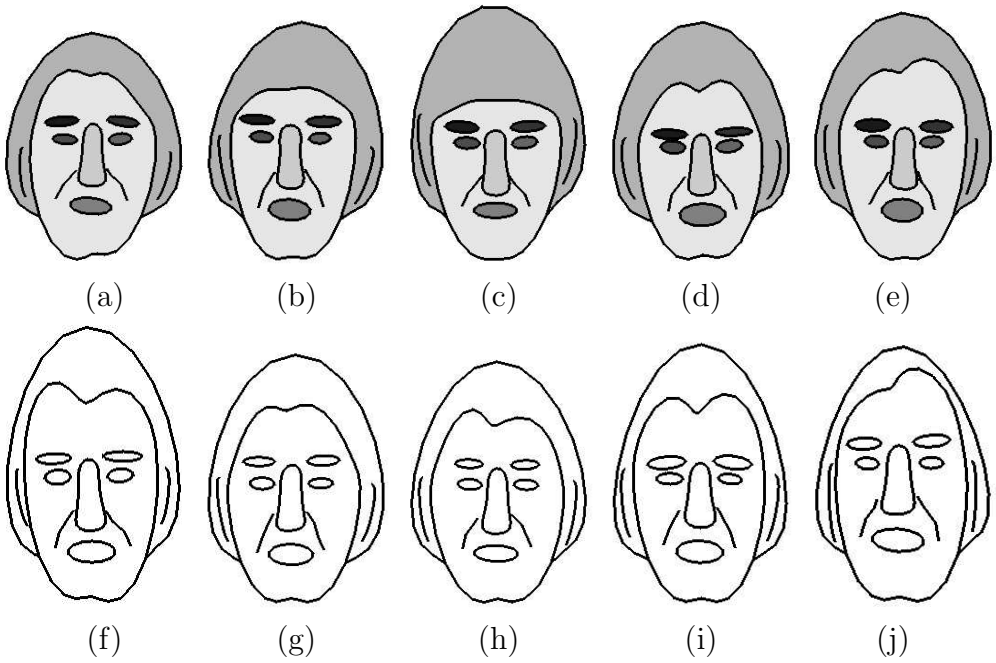


Figure 5.21. Cartoon faces reconstructed from Fourier descriptors using only 30% of the frequency components: (a) to (j) are ten average cartoon faces for ten different subjects based on five images for each subject. Individual components are shown in different shades in (a) to (e).

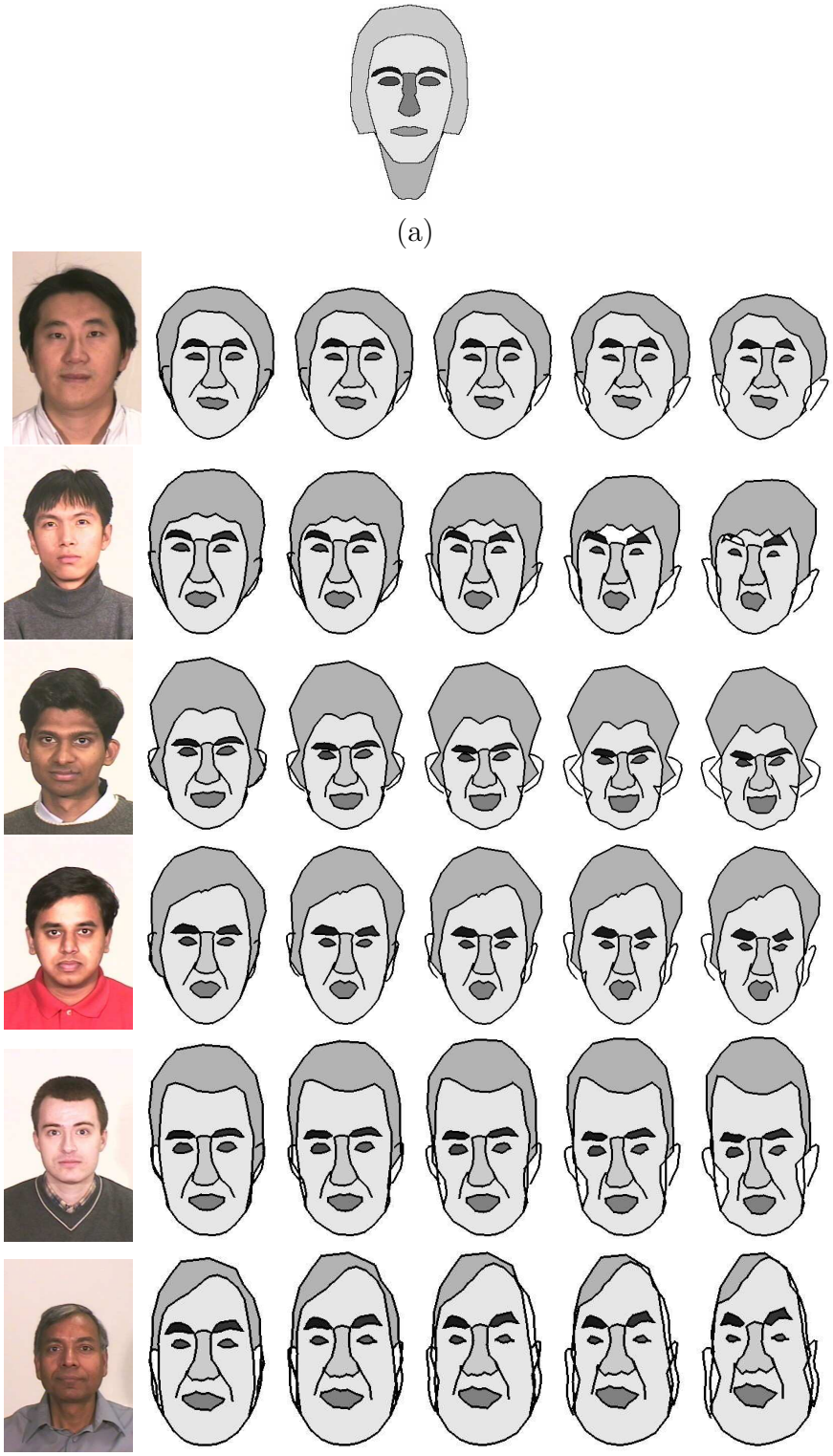
time.

5.5 Facial Caricatures for Recognition and Visualization

Facial caricatures are generated based on exaggeration of an individual's facial distinctiveness from the average facial topology. Let $\mathbf{G_P^{crc}}$ represent the face graphs of caricatures for the subject P, and $\mathbf{G_0}$ be the face graph of the average facial topology. Caricatures are generated via the specification of an exaggeration coefficient, k_i, in Eq. (5.25):

$$\mathbf{G_P^{crc}} = \bigcup_i \mathcal{F}^{-1}\left\{SGD_i^{\mathbf{G_P}} + k_i \cdot \left(SGD_i^{\mathbf{G_P}} - SGD_i^{\mathbf{G_0}}\right)\right\}. \qquad (5.25)$$

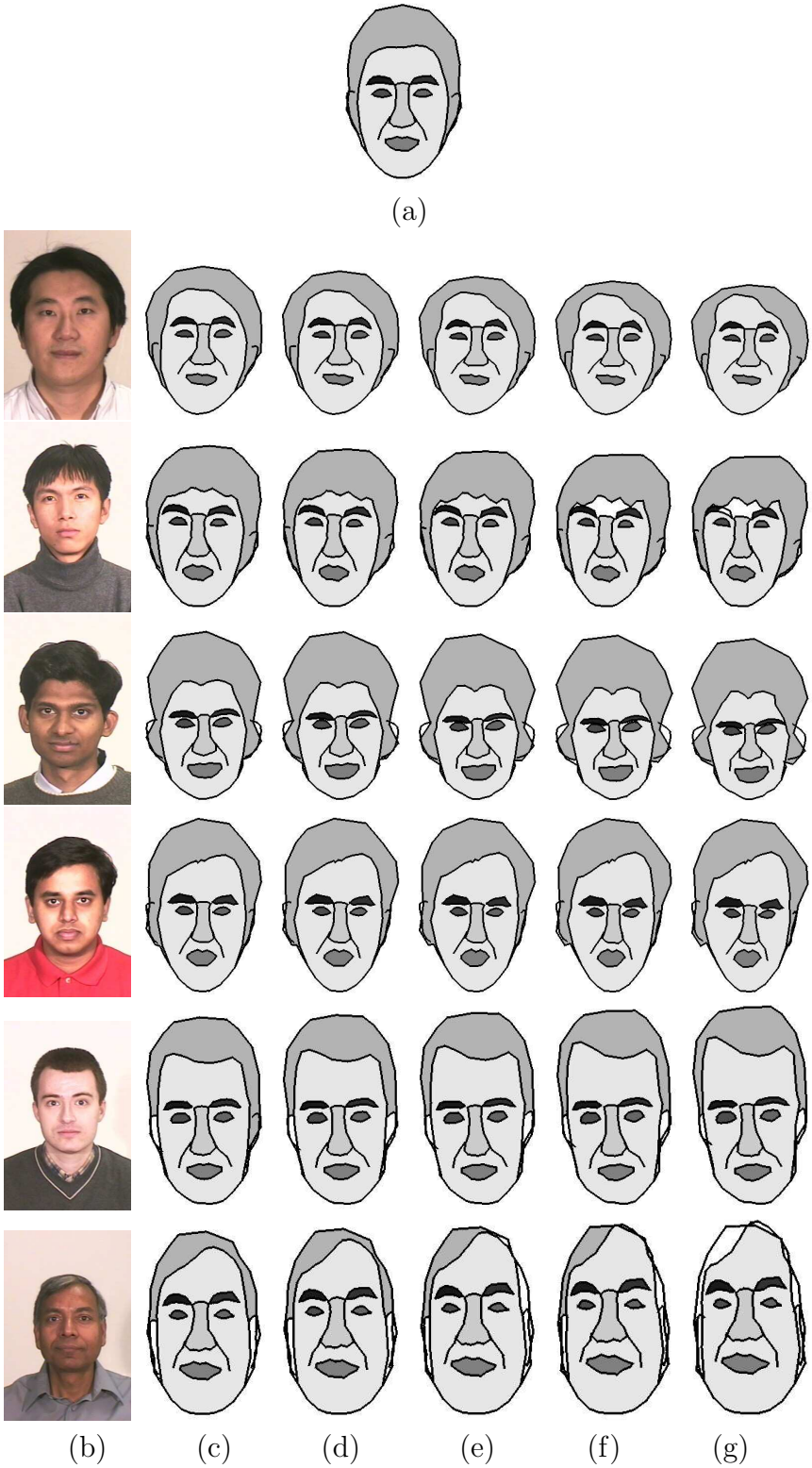
Currently, we use the same value of the coefficient for all the components, i.e., $k_i = k$. Figure 5.22 shows facial caricatures generated with respect to the average facial topology obtained from the 3D generic face model. In Fig. 5.23, facial caricatures are optimized in the sense that the average facial topology is obtained from the mean facial topology of training images (total of 50 images for ten subjects). We can see that it is *easier* for a human to recognize a known face based on the exaggerated faces. We plan to quantitatively evaluate the effect of exaggeration of salient facial features on the performance of a face recognition system. Furthermore, this framework of caricature generation can be easily employed as an alternative to methods of visualizing high dimensional data, e.g., Chernoff faces [184].



(a)

(b) (c) (d) (e) (f) (g)

Figure 5.22. Facial caricatures generated based on a generic 3D face model: (a) a prototype of the semantic face graph, $\mathbf{G_0}$, obtained from a generic 3D face model, with individual components shaded; (b) face images of six different subjects; (c)-(g) caricatures of faces in (b) (semantic face graphs with individual components shown in different shades) with different values of exaggeration coefficients, k, ranging from 0.1 to 0.9.



(a)

(b) (c) (d) (e) (f) (g)

Figure 5.23. Facial caricatures generated based on the average face of 50 faces (5 for each subject):(a) a prototype of the semantic face graph, $\mathbf{G_0}$, obtained from the mean face of the database, with individual components shaded; (b) face images of six different subjects; (c)-(g) caricatures of faces in (b) (semantic face graphs with individual components shown in different shades) with different values of exaggeration coefficients, k, ranging from 0.1 to 0.9.

5.6 Summary

For overcoming variations in pose, illumination, and expression, we propose semantic face graphs that are extracted from a subset of vertices of a 3D face model, and aligned to an image for face recognition. We have presented a framework for semantic face recognition, which is designed to automatically derive weights for facial components based on their distinctiveness and visibility, and to perform face matching based on visible facial components. Face alignment is a crucial module for face matching, and we implement it in a coarse-to-fine fashion. We have shown examples of coarse alignment, and have investigated two deformation approaches for fine alignment of semantic face graphs using interacting snakes. Experimental results show that a successful interaction among multiple snakes associated with facial components makes the semantic face graph a useful model to represent faces (e.g., cartoon faces and caricatures) for recognition.

Our automatic scheme for aligning faces uses interacting snakes for various facial components, including the hair outline, face outline, eyes, nose, and mouth. We are currently adding snakes for eyebrows to completely automate the whole process of face alignment. We plan to test the proposed semantic face matching algorithm on standard face databases. We also plan to implement a pose estimation module based on the alignment results in order to construct an automated pose-invariant face recognition system.

Chapter 6

Conclusions and Future Directions

We will give conclusions and describe future research directions in the following two sections, respectively.

6.1 Conclusions

Face detection as well as recognition are challenging problems and there is still a lot of work that needs to be done in this area. Over the past ten years, face recognition has received substantial attention from researchers in biometrics, pattern recognition, computer vision, and cognitive psychology communities. This common interest in facial recognition technology among researchers working in diverse fields is motivated both by our remarkable ability to recognize people and by the increased attention being devoted to security applications. Applications of face recognition can be found in security, tracking, multimedia, and entertainment domains. We have proposed two paradigms to advance face recognition technology. Three major tasks involved in

such vision-based systems are (i) detection of human faces, (ii) construction of face models/representations for recognition, and (iii) identification of human faces.

Detection of human faces is the first step in our proposed system. It is also the initial step in other applications such as video surveillance, design of human computer interface, face recognition, and face database management. We have proposed a face detection algorithm for color images in the presence of various lighting conditions as well as complex backgrounds. Our detection method first corrects the color bias by a lighting compensation technique that automatically estimates the statistics of reference white for color correction. We overcame the difficulty of detecting the low-luma and high-luma skin tones by applying a nonlinear transformation to the $YCbCr$ color space. Our method detects skin regions over the entire image, and then generates face candidates based on the spatial arrangement of these skin patches. Next, the algorithm constructs eye, mouth, and face boundary maps for verifying each face candidate. Experimental results have demonstrated successful detection of multiple faces of different size, color, position, scale, orientation, 3D pose, and expression in several photo collections.

Construction of face models is closely coupled with recognition of human faces, because the choice of internal representations of human faces greatly affects the design of the face matching or classification algorithm. 3D face models can help augmenting the training face databases used by the appearance-based face recognition approaches to allow for recognition under illumination and head pose variations. For recognition, We have designed two methods for modeling human faces based on a generic 3D face model. One requires individual facial measurements of shape and texture (i.e., color

images with registered range data) captured in the frontal view; the other takes only color images as its facial measurements. Both modeling methods adapt facial features of a generic model to those extracted from an individual's facial measurements in a global-to-local fashion. The first method aligns the model globally, uses the 2.5D active contours to refine feature boundaries, and propagates displacements of model vertices iteratively to smooth non-feature areas. The resulting face model is visually similar to the true face. The resulting 3D model has been shown to be quite useful for recognizing non-frontal views based on an appearance-based recognition algorithm.

The second modeling method aligns semantic facial components, e.g., eyes, mouth, nose, and the face outline, of the generic semantic face graph onto those in a color face image. The nodes of a semantic face graph, derived from a generic 3D face model, represent high-level facial components, and are connected by triangular meshes. The semantic face graph is first coarsely aligned to the locations of detected face and facial components, and then finely adapted to the face image using interacting snakes, each of which describes a semantic component. A successful interaction of these multiple snakes results in appropriate component weights based on distinctiveness and visibility of individual components. Aligned facial components are transformed to a feature space spanned by Fourier descriptors for semantic face matching. The semantic face graph allows face matching based on selected facial components, and updating of a 3D face model based on 2D images. The results of face matching demonstrate the classification and visualization (e.g., the generation of cartoon faces and facial caricatures) of human faces using the derived semantic face graphs.

6.2 Future Directions

Based on the two recognition paradigms proposed and implemented in this thesis, we can extend our work on face detection, modeling and recognition in the following manner:

6.2.1 Face Detection & Tracking

The face detection module can be further improved by

- Optimizing the implementation for real-time applications;
- Combining the global (appearance-based) approach and a modified version of our analytic (feature-based) approach for detecting faces in profile views, in blurred images, and in images captured at a long distance;
- Fusing a head (or body) detector in grayscale images and our skin filter in color images for locating non-skin-tone faces (e.g., faces in gray-scale images or faces taken under extreme lighting conditions).

In order to make the face detection module useful for face tracking, we need to include motion detection and prediction submodules as follows.

- **Parametric face descriptors**: Face ellipses and eye-mouth triangles are useful measurements that can be used for the motion prediction of human faces. We are currently developing a tracking system that combines temporal and shape (derived from our face detection method) information.

- **A face tracking and recognition prototype**: A prototype of a recognition system with tracking modules is shown in Fig. 6.1. The detection and tracking

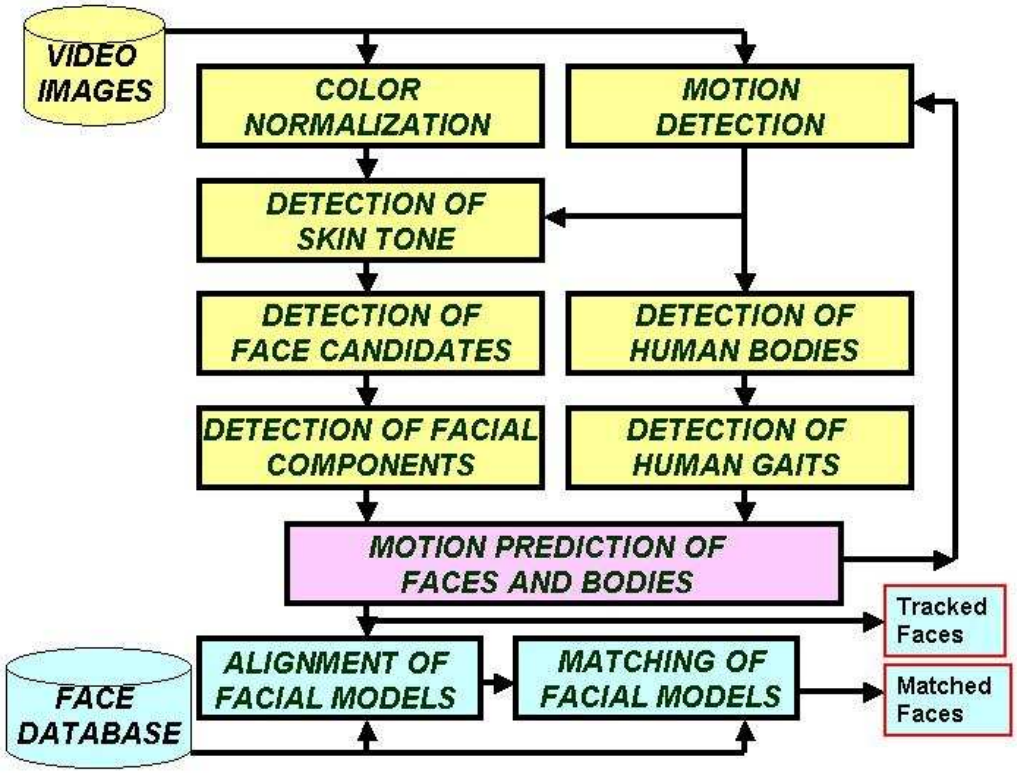
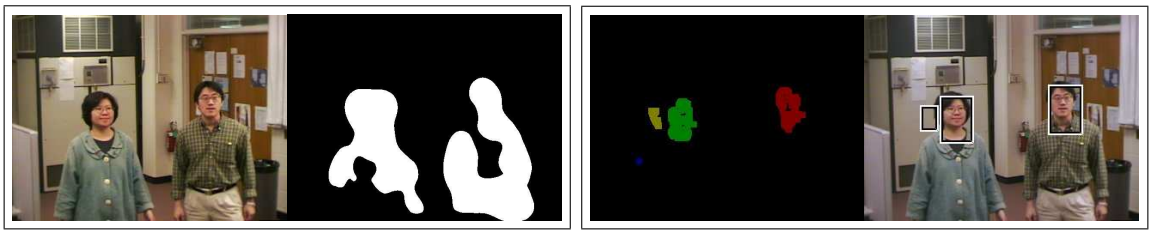


Figure 6.1. A prototype of a face identification system with the tracking function.

modules include (i) a motion filter if temporal information is available, (ii) a human body detector and analysis of human gait, and (iii) a motion predictor of face and the human body.

- **Preliminary tracking results**: An example of detection of motion and skin color is shown in Figure 6.2 (see [185] for more details). The preliminary results of off-line face tracking based on the detection of interframe difference, skin color, and facial features are shown in Fig. 6.3, which contains a sequence of 25 video frames. These images are lighting-compensated and overlaid with detected faces. The image sequence shows two subjects entering the PRIP Lab



(a) (b) (c) (d)

Figure 6.2. An example of motion detection in a video frame: (a) A color video frame; (b) extracted regions with significant motion; (c) detected moving skin patches shown in pseudocolor; (d) extracted face candidates described by rectangles.

in the Engineering building at Michigan State University through two different doors. Faces of different sizes and poses are successfully tracked under various lighting conditions.

6.2.2 Face Modeling

We have developed two face modeling methods for range (with registered color) data and for color data in a frontal view. Once we construct a pose estimator, we can modify the proposed methods of face alignment for non-frontal views. This extension of modeling includes the following tasks:

- **Complete head and ear mesh models**: We need model polygons for hair/head portion and ears in order to generate hair and ear outlines in non-frontal views.

- **Pose and illumination estimation**: We can design a head pose estimator and an illumination estimator for faces in frontal and non-frontal views based on the locations of face and facial components, and shadows/shadings on the face.

Figure 6.3. Face tracking results on a sequence of 25 video frames. These images are arranged from top to bottom and from left to right. Detected faces are overlaid on the lighting-compensated images.

- **Non-frontal training views**: According to the estimated head pose, we can rotate the generic face model and generate the boundary curves of the semantic components for face alignment at the estimated pose.

6.2.3 Face matching

We have designed a semantic face matching algorithm based on the component weights derived from distinctiveness and visibility of individual facial components. Currently, the semantic graph descriptors, SGD_i in Section 5.4.1, used for comparing the difference between facial components contain only the shape information (i.e., component contours). We can improve the performance of the algorithm by including the following properties:

- **Texture information**: Associate a semantic graph descriptor with a set of texture information (e.g., wavelet coefficients, photometric sketches [55], and normalized color values) for each facial component. The semantic face matching algorithm will compare faces based on both the shape and texture information.

- **Scalability**: Evaluate the matching algorithm on several public domain face databases.

- **Caricature effects on recognition**: Explore other weighting functions on the distinctiveness of individual facial components based on the visualized facial caricature and the recognition performance.

- **Facial statistics**: Analyze face shape, race, sex, and age, and construct other

semantic parameters for face recognition, based on a large face database.

Appendices

Appendix A

Transformation of Color Space

In this appendix, we will give the detailed formulae of two types of colorspace transformations and an elliptical skins classifier, which are used in our face detection algorithm. The transformations include a linear transformation between RGB and $YCbCr$ color spaces and a nonlinear transformation applied to $YCbCr$ for compensating the luma dependency. The skin classifier is described by an elliptical region, which lies in the nonlinearly transformed $YCbCr$ space.

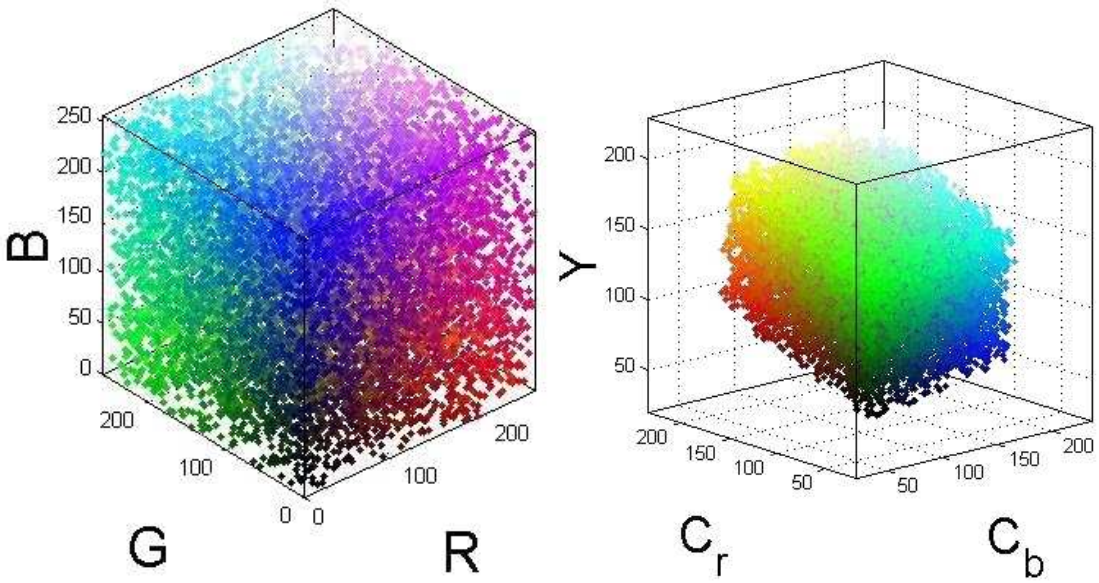
A.1 Linear Transformation

Our face detection algorithm utilizes a linear transformation to convert the color components of an input image in the RGB space into those in the $YCbCr$ space for separating the luma component from chroma components of the input image. The transformation between these two space is formulated in Eqs. (A.1) and (A.2) for the value of color components that range from 0 to 255 (see the details in [155]).

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \frac{1}{256} \begin{bmatrix} 65 & 129.057 & 25.064 \\ -37.945 & -74.494 & 112.439 \\ 112.439 & -94.154 & -18.285 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad \text{(A.1)}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{256} \begin{bmatrix} 298.082 & 0 & 408.583 \\ 298.082 & -100.291 & -208.120 \\ 298.082 & 516.411 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} \quad \text{(A.2)}$$

Figures A.1 (a) and (b) illustrate a set of randomly sampled reproducible colors in the RGB space and its corresponding set in the $YCbCr$ space.



(a) (b)

Figure A.1. Color spaces: (a) RGB; (b) $YCbCr$.

A.2 Nonlinear Transformation

In the YC_bC_r color space, we can regard the chroma (C_b and C_r) as functions of the luma (Y): $C_b(Y)$ and $C_r(Y)$. Let the transformed chroma be $C'_b(Y)$ and $C'_r(Y)$. The nonlinear transformation converts the elongated cluster into a cylinder-like shape,

based on the skin cluster model obtained from a subset of the HHI database. The model is specified by the centers (denoted as $\overline{C_b}(Y)$ and $\overline{C_r}(Y)$) and widths of the cluster (denoted as $Wc_b(Y)$ and $Wc_r(Y)$) (See Fig. 3.5). The following equations describe how this transformation is computed.

$$C_i'(Y) = \begin{cases} \left(C_i(Y) - \overline{C_i}(Y)\right) \cdot \frac{Wc_i}{Wc_i(Y)} + \overline{C_i}(K_h) & \text{if } Y < K_l \text{ or } K_h < Y, \\ C_i(Y) & \text{if } Y \in [K_l, K_h], \end{cases} \tag{A.3}$$

$$Wc_i(Y) = \begin{cases} WLc_i + \frac{(Y - Y_{min}) \cdot (Wc_i - WLc_i)}{K_l - Y_{min}} & \text{if } Y < K_l, \\ WHc_i + \frac{(Y_{max} - Y) \cdot (Wc_i - WHc_i)}{Y_{max} - K_h} & \text{if } K_h < Y, \end{cases} \tag{A.4}$$

$$\overline{C_b}(Y) = \begin{cases} 108 + \frac{(K_l - Y) \cdot (118 - 108)}{K_l - Y_{min}} & \text{if } Y < K_l, \\ 108 + \frac{(Y - K_h) \cdot (118 - 108)}{Y_{max} - K_h} & \text{if } K_h < Y, \end{cases} \tag{A.5}$$

$$\overline{C_r}(Y) = \begin{cases} 154 - \frac{(K_l - Y) \cdot (154 - 144)}{K_l - Y_{min}} & \text{if } Y < K_l, \\ 154 + \frac{(Y - K_h) \cdot (154 - 132)}{Y_{max} - K_h} & \text{if } K_h < Y, \end{cases} \tag{A.6}$$

where C_i in Eqs. (A.3) and (A.4) is either C_b or C_r, $Wc_b = 46.97$, $WLc_b = 23$, $WHc_b = 14$, $Wc_r = 38.76$, $WLc_r = 20$, $WHc_r = 10$, $K_1 = 125$, and $K_h = 188$. All values are estimated from training samples of skin patches on a subset of the HHI images. Y_{min} and Y_{max} in the YC_bC_r color space are 16 and 235, respectively. Note that the boundaries of the cluster are described by two curves $\overline{C_i}(Y) \pm Wc_i(Y)/2$,

and are shown as blue-dashed lines in Fig. 3.5(a) for C_b and in Fig. 3.5(b) for C_r.

A.3 Skin Classifier

The elliptical model for the skin tones in the transformed C'_b-C'_r space is described in Eqs. (A.7) and (A.8), and is depicted in Fig. 3.6.

$$\frac{(x - ecx)^2}{a^2} + \frac{(y - ecy)^2}{b^2} = 1, \tag{A.7}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} C'_b - cx \\ C'_r - cy \end{bmatrix}, \tag{A.8}$$

where $cx = 109.38$, $cy = 152.02$, $\theta = 2.53$ (in radian), $ecx = 1.60$, $exy = 2.41$, $a = 25.39$, and $b = 14.03$. These values are computed from the skin cluster in the C'_b-C'_r space at 1% of outliers.

Appendix B

Distance between Skin Patches

Facial skin areas are usually segmented/split into several clusters of skin patches due to the presence of facial hair, glasses, and shadows. An important issue here is how to group/merge these skin regions based on the spatial distance between them. Since the clusters have irregular shapes, both the Bhattacharrya distance for a generalized Gaussian distribution and the distance based on the circular approximation of the cluster areas do not result in a satisfactory merging. Hence, we combine three types of cluster radii (circular, projection, and elliptical) in order to compute an *effective*

radius of a cluster 'i' w.r.t. another cluster 'j' as follows.

$$R_i = \max(R_i^p, R_i^e) + k \cdot R_i^c \ , \tag{B.1}$$

$$R_i^p = a_i |cos(\theta_{ij})| \ , \tag{B.2}$$

$$R_i^e = \left(\frac{1}{cos^2(\theta_{ij})/a_i^2 + sin^2(\theta_{ij})/b_i^2} \right)^{1/2} \ , \tag{B.3}$$

$$R_i^c = (N_i/\pi)^{1/2} \ , \tag{B.4}$$

where R_i is the effective radius of the cluster i; R_i^p is its projection radius; R_i^e is its elliptical radius; R_i^c is the circular radius used in [84]; the constant k (equals 0.1) is used to prevent the effective radius from vanishing when two clusters are thin and parallel; a_i and b_i are the lengths of the major and minor axes of the cluster i, respectively; θ_{ij} is the angle between the major axis of the cluster i and the segment connecting the centroids of clusters i and j; and N_i is the area of the cluster i. The major and minor axes of the cluster i are estimated by the eigen-decomposition of the covariance matrix

$$C = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} , \tag{B.5}$$

where σ_x, σ_y, and σ_{xy} are the second-order central moments of the skin cluster i. The eigenvalues of the covariance matrix C and the lengths of the major and minor axes

of the cluster are computed by Eqs. (B.6)-(B.10).

$$a_i = \sqrt{\lambda_1} \tag{B.6}$$

$$b_i = \sqrt{\lambda_2} \tag{B.7}$$

$$\lambda_1 = \frac{1}{2} \cdot \left(\sigma_x^2 + \sigma_y^2 + \sqrt{\left(\sigma_x^2 - \sigma_y^2\right)^2 + 4\sigma_{xy}^2} \right), \tag{B.8}$$

$$\lambda_2 = \frac{1}{2} \cdot \left(\sigma_x^2 + \sigma_y^2 - \sqrt{\left(\sigma_x^2 - \sigma_y^2\right)^2 + 4\sigma_{xy}^2} \right), \tag{B.9}$$

$$\alpha = \tan^{-1}\left(\frac{\lambda_1 - \sigma_x^2}{\sigma_{xy}} \right), \tag{B.10}$$

where a_i and b_i are the estimated lengths of the major and minor axes of the cluster, respectively; λ_1 and λ_2 are the largest and smallest eigenvalues of the covariance matrix C, respectively; and α is the orientation of the major axis of the cluster i. The orientation α is used to calculate the angle θ_{ij} in Eq. (B.2). Therefore, the distance between clusters i and j is computed as $d_{ij} = d - R_i - R_j$, where d is the Euclidean distance between the centroids of these two clusters.

Appendix C

Image Processing Template Library (IPTL)

The face detection algorithm has been implemented using **our** Image Processing Template Library (IPTL). This library brings the abstract data class, *Image*, from a class level to a container (a class template) level, *Image Template*. It has the advantages of easy conversion between images of different pixel types, a high reuse rate of image processing algorithms, and a better user interface for manipulating data in the image class level.

C.1 Image and Image Template

An image captures a scene. It is represented by a two-dimensional array of picture elements (so called pixels) in the digital format. Pixels can be of different data types, e.g., one bit for binary images, one byte or word for grayscale images, three bytes

for true-color images, etc. An image is a concrete object; however, *Image* can be regarded as an abstract data class/type in the field of image processing (IP) and computer vision (CV). Contemporary IP libraries, including the Intel IPL [186], the Intel Open CV [187], and the CVIPTool [188], have designed *Image* classes using different pixel depths in bits. Our Image Processing Template Library boosts this abstract *Image* class from a class level to a class template level. The image template, **ImageT**, is designed based on various pixel classes. For example, pixel classes such as *one-bit Boolean*, *one-byte grayscale*, *two-byte grayscale*, and *three-byte true color* are the arguments of the image class template. Hence, the conversion between images of different pixel classes is performed at the pixel level, not at the class level. Hence, a large number of algorithms can be reused for images belonging to different pixel classes.

Figure C.1 shows the architecture of IPTL class templates. The software architecture can be decomposed into five major levels: *platform*, *base*, *pixel*, *image/volume*, and *movie/space* levels. At the platform level, declarations and constants for different working platforms, e.g., the Microsoft Windows and the Sun Unix, are specified in the header file iptl.workingenv.h. At the base level, constants for image processing are defined in the header file iptl.base.h, and space-domain classes for manipulation of different coordinate systems and time-domain classes for evaluation of CPU speed are defined in iptl.geom.h and iptl.time.h, respectively. At the pixel level, the pixel classes such as **GRAY8**, **GRAY16**, and **ColorRGB24** are defined in iptl.pixel.h. At the image level, the image template **ImageT** is defined based on its argument of pixel classes. The IPTL is also designed for a volume template, **VolumeT**, by

considering pixel classes as voxel classes. At the movie level, we can derive templates for color images (e.g., **ImageRGB** and **ImageYCbCr**) for slices, movies, slides, image display, and image analysis based on the image template. Similarly, based on

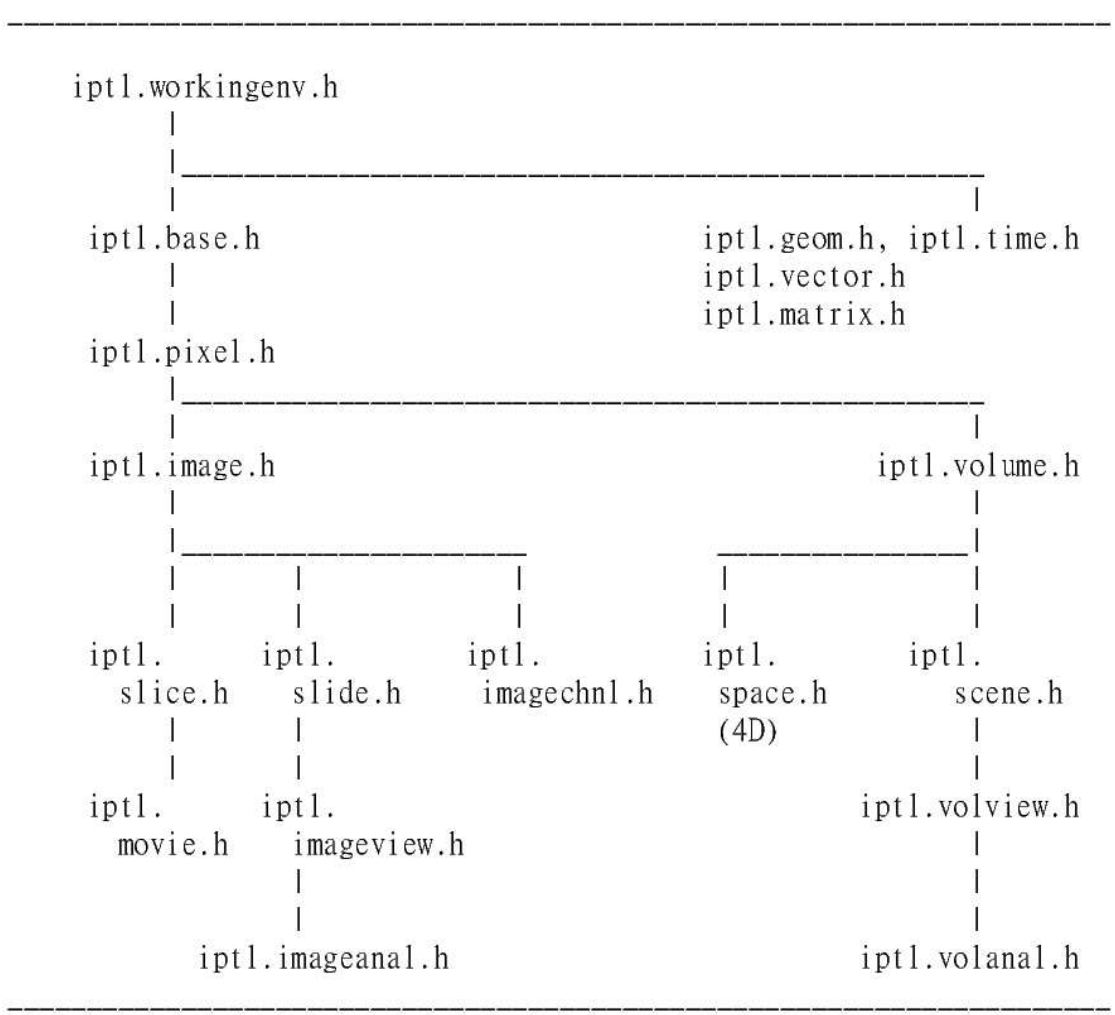


Figure C.1. Architecture of IPTL class templates.

the volume template, we can obtain higher level templates for space, scene, volume display, and volume analysis at the space level.

C.2 Example Code

Example code of the user interface for the pixel classes and the image template is given below.

```
#include "iptl.imagechnl.h"
int main(){

// Conversion of pixel types
//
GREY8 graypix = 10;
FLOAT32 flpix = 3.5;
ColorRGB colorpix(42,53,64);

colorpix = graypix; // Assign a gray value
graypix = flpix; // Truncate data
colorpix.r = 100; // Change the red component
graypix = colorpix; // Compute luminance
flpix += 20.7; // Arithmetic operations

// Image manipulation
//
// Image Creation
ImageT<GREY8> gray8imageA(HOSTRAM, 128, 128, GREY8(55));
ImageT<GREY8> gray8imageB(HOSTRAM, 128, 128, GREY8(100));
ImageT<GREY16> gray16imageC; // An empty image

//Creation of color images
//Data arrangement of the image is RGB... RGB...
ImageT<ColorRGB24> rgbimage(HOSTRAM,32,32,128);

//Data arrangement of the image is RRR...GGG...BBB...
ImageRGB<GREY8> rgbimagechnl(rgbimage1);

// A template function for converting images from one type to another
//
gray8imageA = rgbimage; // Extract Luminance
gray16imageC = gray8imageB; // Enlarge the dynamic range of gray values
gray8imageB -= gray8imageA; // Image subtraction
gray8imageA[5] = 100; // Assess pixels as a 1D vector
```

```
gray8imageA(120,120) = 200; // Assess pixels as a 2D image

} // end of main()
```

We refer the reader to the IPTL reference manual for the details of template implementation.

Bibliography

Bibliography

[1] *Visionics Corporation (FaceIt)*, <http://www.faceit.com/>.

[2] *FaceSnap*, <http://www.person-spotter.de/htdocs/english/frame_right/produkte/facesnap/inhalt.html>.

[3] *Viisage Technology*, <http://www.viisage.com/>.

[4] *Eyematic Interfaces*, <http://www.eyematic.com/>.

[5] *International Biometric Group*, <http://www.biometricgroup.com/>.

[6] R. Hietmeyer, "Biometric identification promises fast and secure processing of airline passengers," *The Int'l Civil Aviation Organization Journal*, vol. 55, no. 9, pp. 10–11, 2000.

[7] R.-L. Hsu and A.K. Jain, "Semantic face matching," *to appear in Proc. IEEE Int'l Conf. Multimedia and Expo (ICME)*, Aug. 2002.

[8] P. Sinha and T. Poggio, "I think I know that face...," *Nature*, vol. 384, pp. 404, Dec. 1996.

[9] *Caricature Zone*, <http://www.magixl.com/heads/view.html>.

[10] *Benjamin Caricature*, <http://www.interchile.com/benjamin>.

[11] *Destiny Prediction*, <http://www.destiny-prediction.com/face/image/pho02.jpg>.

[12] *Marykateandashley.com*, <http://www.marykateandashley.com/>.

[13] *BBC news*, <http://news.bbc.co.uk/hi/english/in_depth/americas/2000/us_elections/profiles/newsid_1012000/1012795.stm>.

[14] *Iransports.net*, <http://iransports.net/olympic/photo_gallery/08.html>.

[15] *MPEG7 Content Set from Heinrich Hertz Institute*, <http://www.darmstadt.gmd.de/mobile/hm/projects/MPEG7/Documents/N2466.html>, Oct. 1998.

[16] *Corbis*, <http://www.corbis.com/>.

[17] *FaceGen, Singular Inversions*, <http://www.facegen.com>.

[18] *The FACEit System*, <http://www.ntu.edu.sg/eee/icis/Staff/yhAng/ faceimages/live_rec.gif>.

[19] R. Féraud, O.J. Bernier, J.-E. Viallet, and M. Collobert, "A fast and accurate face detection based on neural network," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 42–53, Jan. 2001.

[20] D. Maio and D. Maltoni, "Real-time face location on gray-scale static images," *Pattern Recognition*, vol. 33, no. 9, pp. 1525–1539, Sept. 2000.

[21] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Transactions Multimedia*, vol. MM-1, no. 3, pp. 264–277, Sept. 1999.

[22] H. Schneiderman and T. Kanade, "A statistical method for 3D object detection applied to faces and cars," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 746–751, June 2000.

[23] H.A. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 38–44, 1998.

[24] H.A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, Jan. 1998.

[25] K.K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, Jan. 1998.

[26] K.C. Yow and R. Cipolla, "Feature-based human face detection," *Image and Vision Computing*, vol. 25, no. 9, pp. 713–735, Sept. 1997.

[27] M.S. Lew and N. Huijsmans, "Information theory and face detection," *Proc. IEEE Int'l Conf. Pattern Recognition*, pp. 601–605, Aug. 1996.

[28] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[29] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[30] *XM2VTS face database*, <http://xm2vtsdb.ee.surrey.ac.uk/home.html>.

[31] J. Weng and D.L. Swets, "Face recognition," in *Biometrics: Personal Identification in Networked Society*, A.K. Jain, R. Bolle, and S. Pankanti, Eds., pp. 67–86, Kluwer Academic, Boston, MA, 1999.

[32] L. Wiskott, J.M. Fellous, N. Krüger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.

[33] *Reconstruction of Images from Transformed Data & BunchGraph*, <http ://www.cnl.salk.edu/∼wiskott/Projects/{GaborReconstruction.html, BunchGraph.html}>.

[34] P.S. Penev and J.J. Atick, "Local feature analysis: A general statistical theory for object representation," *Network: Computation in Neural Systems*, vol. 7, no. 3, pp. 477–500, 1996.

[35] *Face model generation*, <http://www.cs.rutgers.edu/∼decarlo/anthface.html>.

[36] H. Wechsler, P. Phillips, V. Bruce, F. Soulie, and T. Huang, Eds., *Face Recognition: From Theory to Applications*, Springer-Verlag, 1998.

[37] R. Chellappa, C.L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, pp. 705–740, May 1995.

[38] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips, "Face recognition: A literature survey," *CVL Technical Report, Center for Automation Research, University of Maryland at College Park*, Oct. 2000, <ftp://ftp.cfar.umd.edu/TRs/CVL-Reports-2000/TR4167-zhao.ps.gz>.

[39] *Faceblind.org*, <http://www.faceblind.org/research/index.php>.

[40] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[41] A. Lanitis, C.J. Taylor, and T.F. Cootes, "Toward automatic simulation of aging effects on face images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, Apr. 2002.

[42] A. Yilmaz and M. Gökmen, "Eigenhill vs. eigenface and eigenedge," *Pattern Recognition*, vol. 34, no. 1, pp. 181–184, Jan. 2001.

[43] Y. Adini, Y. Moses, and S. Ullman, "Face recognition: The problem of compensating for changes in illumination direction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 721–732, July 1997.

[44] J.B. Burns, "Recognition via consensus of local moments of brightness and orientation," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 891–898, June 1996.

[45] G.W. Cottrell, M.N. Dailey, C. Padgett, and R. Adolphs, "Is all face processing holistic? the view from UCSD," in *Computational, Geometric, and Process Perspectives on Facial Cognition: Contexts and Challenges*, M. Wenger and J. Townsend, Eds., Lawrence Erlbaum Associates, Mahwah, NJ, 2000.

[46] Li-Fen Chen, Hong-Yuan Mark Liao, Ja-Chen Lin, and Chin-Chuan Han, "Why recognition in a statistics-based face recognition system should be based on the pure face portion: a probabilistic decision-based proof," *Pattern Recognition*, vol. 34, no. 7, pp. 1393–1403, July 2001.

[47] C. Liu and H. Wechsler, "Evolutionary pursuit and its application to face recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 570–582, June 2000.

[48] G. Rhodes and T. Tremewan, "Understanding face recognition: Caricature effects, inversion, and the homogeneity problem," *Visual Cognition*, vol. 1, pp. 257–311, 1994.

[49] V. Bruce and M. Burton, Eds., *Processing Images of Faces*, Ablex publishing, Norwood, NJ, 1992.

[50] M.B. Lewis and R.A. Johnston, "Understanding caricatures of faces," *Quarterly Journal of Experimental Psychology A*, vol. 51, no. 2, pp. 321–346, May 1998.

[51] P.E. Morris and L.H.V. Wickham, "Typicality and face recognition: A critical re-evaluation of the two factor theory," *Quarterly Journal of Experimental Psychology A*, vol. 54, no. 3, pp. 863–877, Aug. 2001.

[52] B. Bates and J. Cleese, Eds., *The Human Face*, Dorling kindersley publishing Inc., New York, NY, 1992.

[53] H. Leder and V. Bruce, "When inverted faces are recognized: The role of configural information in face recognition," *Quarterly Journal of Experimental Psychology A*, vol. 53, no. 2, pp. 513–536, May 2000.

[54] E. Hjelmas and J. Wroldsen, "Recognizing faces from the eyes only," *Proc. 11th Scandinavian Conf. Image Analysis*, June 1999, <http://citeseer.nj.nec.com/hjelmas99recognizing.html>.

[55] R.G. Uhl Jr. and N. da Vitoria Lobo, "A framework for recognizing a facial image from a police sketch," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 586–593, June 1996.

[56] H. Chenand Y.Q. Xu, H.Y. Shum, S.C. Zhu, and N.N. Zhen, "Example-based facial sketch generation with non-parametric sampling," *Proc. IEEE Int'l Conf. Computer Vision*, Vancouver, Canada, July 2001.

[57] S.E. Brennan, "Caricature generator: The dynamic exaggeration of faces by computer," *Leonardo*, vol. 18, no. 3, pp. 170–178, 1985.

[58] R. Mauro and M. Kubovy, "Caricature and face recognition," *Memory & Cognition*, vol. 20, no. 4, 1992.

[59] M. Grudin, "On internal representation in face recognition systems," *Pattern Recognition*, vol. 33, no. 7, pp. 1161–1177, July 2000.

[60] K.M. Lam and H. Yan, "An analytic-to-holistic approach for face recognition based on a single frontal view," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 673–686, July 1998.

[61] D. DeCarlo and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking," *Int'l Journal Computer Vision*, vol. 38, no. 2, pp. 99–127, July 2000.

[62] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 130–136, June 1997.

[63] M.A. Turk and A.P. Pentland, "Face recognition using eigenfaces," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 586–591, June 1991.

[64] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 84–91, June 1994.

[65] R. Brunelli and T. Poggio, "Face recognition: Features vs. templates," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1052, Oct. 1993.

[66] C. Kotropoulos, A. Tefas, and I. Pitas, "Frontal face authentication using morphological elastic graph matching," *IEEE Trans. Image Processing*, vol. 9, pp. 555–560, Apr. 2000.

[67] *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, 2000, <http://www-prima.inrialpes.fr/FG2000/>.

[68] P. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, Mar. 1998.

[69] F.I. Parke and K. Waters, "Appendix 1: Three-dimensional muscle model facial animation," in *Computer Facial Animation*, pp. 337–338, A.K. Peters, 1996, <http://www.crl.research.digital.com/publications/books/waters/ waters_book. html>.

[70] G. Miller, E. Hoffert, S.E. Chen, E. Patterson, D. Blackketter, S. Rubin, S. A. Applin, D. Yim, and J. Hanan, "The virtual museum: Interactive 3D navigation of a multimedia database," *The Journal of Visualization and Computer Animation*, vol. 3, no. 3, pp. 183–197, July 1992.

[71] H. Fuchs, G. Bishop, K. Arthur, L. McMillan, R. Bajcsy, S. Lee, H. Farid, and T. Kanade, "Virtual space teleconferencing using a sea of cameras," *Technical Report, Department of Computer Science, University of North Carolina-Chapel Hill, Number TR94-033*, May 1994.

[72] M.J.T. Reinders, P.J.L. van Beek, B. Sankur, and J.C.A. van der Lubbe, "Facial feature localization and adaptation of a generic face model for model-based coding," *Signal Processing: Image Communication*, vol. 7, no. 1, pp. 57–74, Mar. 1995, <http://www-it.et.tudelft.nl/itbibliography/reports/1995/journal/imagecom95.reinders.ps.gz>.

[73] A. Hilton, D. Beresford, T. Gentils, R. Smith, and W. Sun, "Virtual people: Capturing human models to populate virtual worlds," *Proc. IEEE Conf. Computer Animation*, pp. 174–185, May 1999.

[74] R.T. Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, Aug. 1997.

[75] G. Taubin and J. Rossignac, "Geometric compression through topological surgery," *ACM Trans. Graphics*, vol. 17, pp. 84–115, Apr. 1998.

[76] M. Lounsbery, T.D. DeRose, and J. Warren, "Multiresolution analysis for surfaces of arbitrary topological type," *ACM Trans. Graphics*, vol. 16, pp. 34–73, Jan. 1997.

[77] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *Int'l Journal Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[78] W.-S. Hwang and J. Weng, "Hierarchical discriminant regression," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1277–1293, Nov. 2000.

[79] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," *Science*, vol. 291, no. 5504, pp. 599–600, Jan. 2000.

[80] J. Weng, C.H. Evans, and W.S. Hwang, "An incremental learning method for face recognition under continuous video stream," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 251–256, Mar. 2000.

[81] M. Pantic and L.J.M. Rothkrantz, "Automatic analysis of facial expressions: The state of the art," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1424–1445, Dec. 1996.

[82] M.-H. Yang, N. Ahuja, and D. Kriegman, "Detecting faces in images: A survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, Jan. 2001.

[83] E. Hjelm and B.K. Low, "Face detection: A survey," *Computer Vision and Image Understanding*, vol. 83, pp. 236–274, Sept. 2001.

[84] M. Abdel-Mottaleb and A. Elgammal, "Face detection in complex environments from color images," *Proc. IEEE Int'l Conf. Image Processing*, pp. 622–626, 1999.

[85] H. Wu, Q. Chen, and M. Yachida, "Face detection from color images using a fuzzy pattern matching method," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, pp. 557–563, June 1999.

[86] A. Colmenarez, B. Frey, and T. Huang, "Detection and tracking of faces and facial features," *Proc. IEEE Int'l Conf. Image Processing*, pp. 657–661, Oct. 1999.

[87] S.C. Dass and A.K. Jain, "Markov face models," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 680–687, July 2001.

[88] A.J. Colmenarez and T.S. Huang, "Face detection with information based maximum discrimination," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 782–787, June 1997.

[89] V. Bakic and G. Stockman, "Menu selection by facial aspect," *Proc. Vision Interface, Canada*, pp. 203–209, May 1999.

[90] K. Sobottka and I. Pitas, "A novel method for automatic face segmentation, facial feature extraction and tracking," *Signal Processing: Image Communication*, vol. 12, no. 3, pp. 263–281, June 1998.

[91] H.D. Ellis, M.A. Jeeves, F. Newcombe, and A. Young, Eds., *Aspects of Face Processing*, Martinus Nijhoff Publishers, Dordrecht, Netherlands, 1985.

[92] O.A. Uwechue and A.S. Pandya, Eds., *Human Face Recognition Using Third-order Synthetic Neural Networks*, Kluwer Academic Publishers, Norwell, MA, 1997.

[93] P.L. Hallinan, G.G. Gordon, A.L. Yuille, P. Giblin, and D. Mumford, *Two- and Three- Dimensional Patterns of the Face*, A.K. Peters, Natick, MA, 1999.

[94] S. Gong, S.J. McKenna, and A. Psarrou, *Dynamic Vision: From Images to Face Recognition*, Imperial College Press, London, 1999.

[95] *MIT face database*, <ftp://whitechapel.media.mit.edu/pub/images/>.

[96] *Yale face database*, <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>.

[97] *AR face database*, <http://rvl1.ecn.purdue.edu/∼aleix/aleix_face_DB.html>.

[98] *Olivetti face database*, <http://www.cam-orl.co.uk/facedatabase.html>.

[99] B. Moghaddam and A. Pentland, "Face recognition using view-based and modular eigenspaces," *Automatic Systems for the Identification and Inspection of Humans, Proc. SPIE*, vol. 2257, pp. 12–21, July 1994.

[100] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, July 1997.

[101] P.S. Penev and L. Sirovich, "The global dimensionality of face space," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 264–270, Mar. 2000.

[102] I.J. Cox, J. Ghosn, and P.N. Yianilos, "Feature-based face recognition using mixture-distance," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 209–216, June 1996.

[103] M.S. Kamel, H.C. Shen, A.K.C. Wong, and R. I. Campeanu, "System for the recognition of human faces," *IBM Systems Journal*, vol. 32, no. 2, pp. 307–320, 1993.

[104] I. Craw, N. Costen, T. Kato, G. Robertson, and S. Akamatsu, "Automatic face recognition: Combining configuration and texture," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 53–58, Sept. 1995.

[105] G.G. Gordon and M. E. Lewis, "Face recognition using video clips and mug shots," *Proc. Office of National Drug Control Policy (ONDCP) Int'l Technical Symposium*, Oct. 1995, <http://http://www.vincent-net.com/gaile/papers/ONDCPPaper/ONDCPPaper.html>.

[106] R. Lengagne, J.-P. Tarel, and O. Monga, "From 2D images to 3D face geometry," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 301–306, Oct. 1996.

[107] J.J. Atick, P.A. Griffin, and A.N. Redlich, "Statistical approach to shape from shading: Reconstruction of 3D face surfaces from single 2D images," *Neural Computation*, vol. 8, no. 6, pp. 1321–1340, Aug. 1996.

[108] Y. Yan and J. Zhang, "Rotation-invariant 3D recognition for face recognition," *Proc. IEEE Int'l Conf. Image Processing*, vol. 1, pp. 156–160, Oct. 1998.

[109] W.Y. Zhao and R. Chellappa, "3D model enhanced face recognition," *Proc. IEEE Int'l Conf. Image Processing*, Sept. 2000, <http://www.cfar.umd.edu/~wyzhao/icip003Dmodel.ps>.

[110] P. Sinha and T. Poggio, "Role of learning in three-dimensional form perception," *Nature*, vol. 384, pp. 460–463, Dec. 1996.

[111] D. DeCarlo, D. Metaxas, and M. Stone, "An anthropometric face model using variational techniques," *Proc. SIGGRAPH Conf.*, pp. 67–74, July 1998.

[112] Q. Chen and G. Medioni, "Building human face models from two images," *IEEE 2nd Workshop on Multimedia Signal Processing*, pp. 117–122, Dec. 1998.

[113] B. Kim and P. Burger, "Depth and shape from shading using the photometric stereo method," *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 54, no. 3, pp. 416–427, 1991.

[114] D.R. Hougen and N. Ahuja, "Estimation of the light source distribution and its use in integrated shape recovery from stereo and shading," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 148–155, May 1993.

[115] J. Cryer, P.-S. Tsai, and M. Shah, "Integration of shape from shading and stereo," *Pattern Recognition*, vol. 28, no. 7, pp. 1033–1043, July 1995.

[116] F.I. Parke and K. Waters, "Modeling faces," in *Computer Facial Animation*, pp. 55–104, A.K. Peters, Wellesley, MA, 1996.

[117] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin, "Making faces," *Proc. SIGGRAPH Conf.*, pp. 55–66, July 1998.

[118] L. Yin and A. Basu, "MPEG4 face modeling using fiducial points," *Proc. IEEE Int'l Conf. Image Processing*, pp. 109–112, Oct. 1997.

[119] W. Lee and N. Magnenat-Thalmann, "Fast head modeling for animation," *Image and Vision Computing*, vol. 18, no. 4, pp. 355–364, Mar. 2000.

[120] R. Lengagne, P. Fua, and O. Monga, "3D stereo reconstruction of human faces driven by differential constraints," *Image and Vision Computing*, vol. 18, no. 4, pp. 337–343, Mar. 2000.

[121] P. Fua, "Using model-driven bundle-adjustment to model heads from raw video sequences," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 46–53, Sept. 1999.

[122] J. Ahlberg, "An experiment on 3D face model adaptation using the active appearance algorithm," *Technical Report, Dept. of EE, Linköping University, Sweden*, Jan. 2001.

[123] J. Ahlberg, "Candide-3 - an updated parameterized face," *Report No. LiTH-ISY-R-2326, Dept. of EE, Linköping University, Sweden*, Jan. 2001.

[124] A.A. Amini, T.E. Weymouth, and R.C. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 9, pp. 855–867, Sept. 1990.

[125] B. Olstad and A.H. Torp, "Encoding of a priori information in active contour models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 863–872, Sept. 1996.

[126] L.H. Staib and J.S. Duncan, "Boundary finding with parametrically deformable models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1061–1075, Nov. 1992.

[127] C.Y. Xu and J.L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 359–369, Mar. 1998.

[128] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, "Fast geodesic active contours," *IEEE Trans. Image Processing*, vol. 10, no. 10, pp. 1467–1475, Oct. 2001.

[129] X.M. Pardo, M.J. Carreira, A. Mosquera, and D. Cabello, "A snake for CT image segmentation integrating region and edge information," *Image and Vision Computing*, vol. 19, no. 7, pp. 461–475, May 2001.

[130] T.F. Chan and L.A. Vese, "Active contours without edges," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 266–277, Feb. 2001.

[131] C. Chesnaud, P. Refregier, and V. Boulet, "Statistical region snake-based segmentation adapted to different physical noise models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1145–1157, Nov. 1999.

[132] S.C. Zhu and A. Yuille, "Region competition - unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, Sept. 1996.

[133] J. Ivins and J. Porrill, "Statistical snakes: active region models," *Proc. Fifth British Machine Vision Conf. (BMVC)*, vol. 2, pp. 377–386, Dec. 1994.

[134] T. Abe and Y. Matsuzawa, "Multiple active contour models with application to region extraction," *Proc. 15th Int'l Conf. Pattern Recognition*, vol. 1, pp. 626–630, Sept. 2000.

[135] V. Chalana, D.T. Linker, D.R. Haynor, and Y.M. Kim, "A multiple active contour model for cardiac boundary detection on echocardiographic sequences," *IEEE Trans. Medical Imaging*, vol. 15, no. 3, pp. 290–298, 1996.

[136] B. Fleming, *3D Modeling & Surfacing*, Morgan Kaufmann, San Francisco, California, 1999.

[137] M. Deering, "Geometry compression," *Proc. SIGGRAPH Conf.*, pp. 13–20, Aug. 1995.

[138] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes," *Proc. SIGGRAPH Conf.*, pp. 173–182, Aug. 1995.

[139] R.-L. Hsu, A.K. Jain, and M. Tuceryan, "Multiresolution model compression using 3-D wavelets," *Proc. Asian Conf. Computer Vision*, pp. 74–79, Jan. 2000.

[140] A.R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Lossless image compression using integer to integer wavelet transforms," *Proc. IEEE Int'l Conf. Image Processing*, vol. 1, pp. 596–599, Oct. 1997.

[141] M.D. Adams and F. Kossentini, "Reversible integer-to-integer wavelet transforms for image compression: Performance evaluation and analysis," *Proc. IEEE Int'l Conf. Image Processing*, vol. 9, no. 6, pp. 1010–1024, June 2000.

[142] *IBM Query By Image Content (QBIC)*, <http://wwwqbic.almaden.ibm.com/>.

[143] *Photobook*, <http://www-white.media.mit.edu/vismod/demos/photobook/>.

[144] M. Abdel-Mottaleb, N. Dimitrova, R. Desai, and J. Martino, "Conivas: Content-based image and video access system," *Proc. Fourth ACM Multimedia Conf.*, pp. 427–428, Nov. 1996.

[145] *FourEyes*, <http://www-white.media.mit.edu/vismod/demos/photobook/foureyes/>.

[146] *Virage*, <http://www.virage.com/>.

[147] J.-Y. Chen, C. Taskiran, E.J. Delp, and C.A. Bouman, "Vibe: A new paradigm for video database browsing and search," *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries*, pp. 96–100, June 1998, <http://stargate.ecn.purdue.edu/∼ips/ViBE/>.

[148] S.-F. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong, "An automated content-based video search system using visual cues," *Proc. ACM Multimedia*, vol. 18, no. 1, pp. 313–324, Nov. 1997, <http://www.ctr.columbia.edu/VideoQ/>.

[149] *Visualseek*, <http://www.ctr.columbia.edu/visualseek/>.

[150] W.Y. Ma and B.S. Manjunath, "Netra: A toolbox for navigating large image databases," *Proc. IEEE Int'l Conf. Image Processing*, vol. 1, pp. 568–571, Oct. 1997.

[151] S. Mehrotra, Y. Rui, M. Ortega, and T.S. Huang, "Supporting content-based queries over images in MARS," *Proc. IEEE Int'l Conf. Multimedia Computing and Systems*, pp. 632–633, June 1997.

[152] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja, "Picsom - content-based image retrieval with self-organizing maps," *Pattern Recognition Letters*, vol. 21, no. 13-14, pp. 1199–1207, Dec. 2000, <http://www.cis.hut.fi/picsom/>.

[153] M.S. Lew, "Next-generation web searches for visual content," *IEEE Computer*, pp. 46–53, Nov. 2000, <http://skynet.liacs.nl>.

[154] A. Vailaya, M. Figueiredo, A.K. Jain, and H.-J. Zhang, "Image classification for content-based indexing," *IEEE Trans. Image Processing*, vol. 10, no. 1, pp. 117–130, Jan. 2001.

[155] C.A. Poynton, *A Technical Introduction to Digital Video*, John Wiley & Sons, New York, 1996.

[156] L.M. Bergasa, M. Mazo, A. Gardel, M.A. Sotelo, and L. Boquet, "Unsupervised and adaptive gaussian skin-color model," *Image and Vision Computing*, vol. 18, pp. 987–1003, Sept. 2000.

[157] J.C. Terrillon, M.N. Shirazi, H. Fukamachi, and S. Akamatsu, "Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 54–61, Mar. 2000.

[158] E. Saber and A.M. Tekalp, "Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions," *Pattern Recognition Letters*, vol. 19, no. 8, pp. 669–680, June 1998.

[159] M.J. Jones and J.M. Rehg, "Statistical color models with application to skin detection," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 274–280, June 1999.

[160] B. Menser and M. Brunig, "Locating human faces in color images with complex background," *Intelligent Signal Processing and Communications Systems*, pp. 533–536, Dec. 1999.

[161] T. Horprasert, Y. Yacoob, and L.S. Davis, "Computing 3-D head orientation from a monocular image," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 242–247, Oct. 1996.

[162] A. Nikolaidis and I. Pitas, "Facial feature extraction and determination of pose," *Pattern Recognition*, vol. 33, pp. 1783–1791, 2000.

[163] W. Huang, Q. Sun, C.-P. Lam, and J.-K. Wu, "A robust approach to face and eyes detection from images with cluttered background," *Proc. IEEE Int'l Conf. Pattern Recognition*, vol. 1, pp. 110–114, Aug. 1998.

[164] K.M. Lam and H. Yan, "Locating and extracting the eye in human face images," *Pattern Recognition*, vol. 29, no. 5, pp. 771–779, 1996.

[165] A. Lanitis, C.J. Taylor, and T.F. Cootes, "Automatic interpretation and coding of face images using flexible models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 743–756, July 1997.

[166] S. Sirohey and A. Rosenfeld, "Eye detection," *Technical Report CS-TR-3971, Univ. of Maryland*, Dec. 1998.

[167] F. Smeraldi, O. Carmona, and J. Bigün, "Saccadic search with Gabor features applied to eye detection and real-time head tracking," *Image and Vision Computing*, vol. 18, no. 4, pp. 323–329, 2000.

[168] D. Stork and M Henneke, "Speechreading: An overview of image processing, feature extraction, sensory integration and pattern recognition techniques," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. xvi–xxvi, 1996.

[169] P.T. Jackway and M. Deriche, "Scale-space properties of the multiscale morphological dilation-erosion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, pp. 38–51, Jan. 1996.

[170] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, Nov. 1986.

[171] *The Champion Database*, <http://www.libfind.unl.edu/alumni/events/ breakfast_for_champions.htm>, Mar. 2001.

[172] *Yahoo news photos*, <http://dailynews.yahoo.com/h/g/ts/>, Dec. 2001.

[173] R.-L. Hsu and A.K. Jain, "Face modeling for recognition," *Proc. IEEE Int'l Conf. Image Processing*, vol. 2, pp. 693–696, Oct. 2001.

[174] *SAMPL range databases*, <http://sampl.eng.ohio-state.edu/∼sampl/data/3DDB/RID/minolta/faceimages.0300/>.

[175] R.-L. Hsu, M. Abdel-Mottaleb, and A.K. Jain, "Face detection in color images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696–706, May 2002.

[176] D. Terzopoulos and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 569–579, 1993.

[177] D. Reisfeld, H. Wolfson, , and Y. Yeshurun, "Contex free attentional operators: The generalized symmetry transform," *Int'l Journal Computer Vision*, vol. 14, pp. 119–130, 1995.

[178] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman, "Linear time euclidean distance transform algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 529–533, May 1995.

[179] R. Zhang, P.-S. Tsai, J. Cryer, and M. Shah, "Shape from shading: a survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 690–706, Aug. 1999.

[180] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1999.

[181] V. Caselles, R. Kimmel, , and G. Sapiro, "Geodesic active contours," *Int'l Journal Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[182] D. Adalsteinsson and J.A. Sethian, "A fast level set method for propagating interfaces," *Journal Computational Physics*, vol. 118, pp. 269–277, 1995.

[183] X. Han, C. Xu, and J.L. Prince, "A topology preserving deformable model using level sets," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, vol. II, pp. 765–770, Dec. 2001.

[184] H. Chernoff, "The use of faces to represent points in k-dimensional space graphically," *Journal of the Americal Statistics Association*, vol. 68, pp. 361–368, 1973.

[185] R.-L. Hsu and A.K. Jain, "Detection and tracking of multiple faces in video," *Technical Report MSU-CSE-02-13*, vol. Dept. Computer Science & Engineering, pp. Michigan State University, May 2002.

[186] *Intel Image Processing Library*, <http://developer.intel.com/software/products /perflib/ipl/index.htm>.

[187] *Intel Open Source Computer Vision Library*, <http://developer.intel.com/ software/opensource/cvfl/opencv_download.htm>.

[188] S.E. Umbaugh, *Computer Vision and Image Processing: A Practical Approach Using CVIPTools*, Prentice Hall, Upper Saddle River, NJ, 1997.